



The following document contains information on Cypress products. Although the document is marked with the name "Spansion" and "Fujitsu", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

Continuity of Specifications

There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

Continuity of Ordering Part Numbers

Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress (NASDAQ: CY) delivers high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. With a broad, differentiated product portfolio that includes NOR flash memories, F-RAM™ and SRAM, Traveo™ microcontrollers, the industry's only PSoC® programmable system-on-chip solutions, analog and PMIC Power Management ICs, CapSense® capacitive touch-sensing controllers, and Wireless BLE Bluetooth® Low-Energy and USB connectivity solutions, Cypress is committed to providing its customers worldwide with consistent innovation, best-in-class support and exceptional system value.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spancion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spancion product under development by Spancion. Spancion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spancion assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2013 Spancion Inc. All rights reserved. Spancion®, the Spancion logo, MirrorBit®, MirrorBit® Eclipse™, ORNAND™ and combinations thereof, are trademarks and registered trademarks of Spancion LLC in the United States and other countries. Other names used are for informational purposes only and may be trademarks of their respective owners.

FR60
32-BIT MICROCONTROLLER
MB91460 Series
HARDWARE MANUAL
Version 3.0

FR60

32-BIT MICROCONTROLLER

MB91460 Series

Hardware Manual

- The contents of this document are subject to change without notice.
- Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU SEMICONDUCTOR device; FUJITSU SEMICONDUCTOR does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information.
FUJITSU SEMICONDUCTOR assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU SEMICONDUCTOR or any third party or does FUJITSU SEMICONDUCTOR warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU SEMICONDUCTOR assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
- Please note that FUJITSU SEMICONDUCTOR will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

©2011 FUJITSU SEMICONDUCTOR EUROPE GmbH

Table of Contents

Chapter 1	Introduction, Handling and Precautions	1
1.1.	Introduction	1
1.2.	Handling Precautions.....	2
1.3.	Software Precautions.....	7
1.4.	Notes on Debugger.....	10
1.5.	How to Use This Document	11
Chapter 2	MB91460 Rev.A/Rev.B Overview	17
2.1.	Overview	17
2.2.	Features.....	17
2.3.	MB91460 Series Product Lineup	25
2.4.	Block Diagram	30
2.5.	Debug Support Features	31
Chapter 3	MB91460 Series Basic Information	35
3.1.	Memory Map.....	35
3.2.	I/O Map	36
3.3.	Interrupt Vector Table	113
3.4.	Package	121
3.5.	Pin Assignment Diagram	123
3.6.	Pin Definitions MB91V460A.....	126
3.7.	I/O Circuit Types MB91V460A.....	141
3.8.	Pin Definitions MB91FV460B	143
3.9.	I/O Circuit Types MB91FV460B.....	179
3.10.	Pin State Table	192
Chapter 4	CPU Architecture	201
4.1.	Overview	201
4.2.	Features.....	202
4.3.	CPU	203
4.4.	32-bit/16-bit Bus Converter	203
4.5.	Harvard/Princeton Bus Converter.....	203
4.6.	Instruction Overview	204
4.7.	Data Structure.....	205
4.8.	Word Alignment	206
4.9.	Addressing.....	207
Chapter 5	CPU Registers	209
5.1.	General-purpose Registers.....	209
5.2.	Dedicated Registers	209

Chapter 6	EIT: Exceptions, Interrupts and Traps	217
6.1.	Overview	217
6.2.	Features.....	217
6.3.	EIT Trigger	217
6.4.	Context saving	218
6.5.	Recovery from EIT handler	218
6.6.	EIT Interrupt Level	219
6.7.	EIT Vector Table	219
6.8.	Multiple EIT Processing	220
6.9.	Operation	222
6.10.	Caution	225
Chapter 7	Branch Instruction	227
7.1.	Overview	227
7.2.	List of Delayed Branching instructions.....	227
7.3.	Operation of Delayed Branching Instructions	227
7.4.	Some Examples of Delayed Branching Instructions.....	228
7.5.	Restrictions on Branch Instruction with Delay Slot	229
7.6.	Branch Instruction without Delay Slot	230
7.7.	List of Non-Delayed Branch Instructions	230
Chapter 8	Device State Transition	231
8.1.	Overview	231
8.2.	Features.....	231
8.3.	State Transition Diagram	232
Chapter 9	Reset	237
9.1.	Overview	237
9.2.	Features.....	237
9.3.	Configuration	239
9.4.	Registers.....	240
9.5.	INITX Pin Input (INIT: Settings Initialization Reset)	245
9.6.	Hardware Watchdog / Clock Supervisor reset (INIT: Settings Initialization Reset)	247
9.7.	Software Watchdog Reset (INIT: Settings Initialization Reset).....	248
9.8.	Software Reset (RST: Operation Initialization Reset).....	249
9.9.	Reset Operation Modes	250
9.10.	MCU Operation Mode	251
9.11.	Caution	252
Chapter 10	Standby	253
10.1.	Overview	253
10.2.	Features.....	253
10.3.	Configuration	254
10.4.	Registers.....	255
10.5.	Operation	257
10.6.	Settings.....	259

MB91460 Series

10.7. Q&A	259
10.8. Caution	262
Chapter 11 Memory Controller, Flash and F-Cache	263
11.1. Overview	263
11.2. Main Flash Interface	263
11.3. Embedded ID-RAM	263
11.4. Flash Cache and Data Buffer	263
11.5. Prefetch	263
11.6. Fixed Mode and Reset Vectors	264
11.7. Registers	264
11.8. Explanations of Registers	265
Chapter 12 External Bus Instruction Cache (I-Cache).....	275
12.1. Overview	275
12.2. Main Unit Structure	275
12.3. Tag Structure	276
12.4. Register	277
12.5. Address Map	278
12.6. Operating Modes	284
12.7. Setting Method When I-cache is Used	286
Chapter 13 Clock Control.....	289
13.1. Overview	289
13.2. Features	289
13.3. Configuration	291
13.4. Registers	292
13.5. Operation	304
13.6. Settings	307
13.7. Q & A	308
13.8. Caution	311
Chapter 14 PLL Interfaces (Main PLL, E-Ray, USB)	313
14.1. Overview	313
14.2. Main PLL Interface	313
14.3. E-Ray and USB PLL Interfaces	315
14.4. Registers	317
14.5. PLL Interface Interrupts	324
14.6. Recommended Settings	325
14.7. Caution	327
14.8. Procedure example for Main PLL	328
Chapter 15 CAN Clock Prescaler	329
15.1. Overview	329
15.2. Features	329
15.3. Registers	330

Chapter 16 Clock Supervisor.....	333
16.1. Overview Clock Supervisor.....	333
16.2. Clock Supervisor Register	335
16.3. Block Diagram Clock Supervisor	337
16.4. Operation Modes	338
16.5. Cautions.....	354
Chapter 17 Clock Modulator	355
17.1. Overview	355
17.2. Clock Modulator Registers.....	356
17.3. Application Note.....	362
Chapter 18 Timebase Counter.....	363
18.1. Overview	363
18.2. Features.....	364
18.3. Configuration	365
18.4. Registers.....	366
18.5. Operation	369
18.6. Settings.....	375
18.7. Q&A	376
18.8. Caution	378
Chapter 19 Timebase Timer.....	379
19.1. Overview	379
19.2. Features.....	379
19.3. Configuration	380
19.4. Register	381
19.5. Operation	383
19.6. Setting.....	384
19.7. Q & A	385
19.8. Caution	387
Chapter 20 Software Watchdog Timer.....	389
20.1. Overview	389
20.2. Features.....	389
20.3. Configuration	390
20.4. Register	391
20.5. Operation	394
20.6. Setting.....	396
20.7. Q & A	397
20.8. Caution	398
Chapter 21 Hardware Watchdog Timer.....	399
21.1. Overview	399
21.2. Register	401

MB91460 Series

21.3. Functions	403
21.4. Caution	404
Chapter 22 Main Oscillation Stabilization Timer.....	405
22.1. Overview	405
22.2. Features.....	405
22.3. Configuration	406
22.4. Register	407
22.5. Operation	409
22.6. Setting.....	411
22.7. Q & A	412
22.8. Caution	414
Chapter 23 Sub Oscillation Stabilization Timer	417
23.1. Overview	417
23.2. Features.....	417
23.3. Configuration	418
23.4. Register	419
23.5. Operation	420
23.6. Setting.....	423
23.7. Q & A	424
23.8. Caution	426
Chapter 24 Interrupt Control.....	429
24.1. Overview	429
24.2. Features.....	429
24.3. Configuration	430
24.4. Registers.....	431
24.5. Operation	435
24.6. Setting.....	438
24.7. Q & A	438
24.8. Caution	439
Chapter 25 External Interrupt and NMI	441
25.1. Overview	441
25.2. Features.....	441
25.3. Configuration	442
25.4. Control/Status Register Description.....	450
25.5. Operation	457
25.6. Setting.....	460
25.7. Q & A	460
25.8. Caution	464
Chapter 26 DMA Controller	465
26.1. Overview of the DMA Controller (DMAC)	465
26.2. DMA Controller (DMAC) Registers	467

26.3. DMA Controller (DMAC) Operation	489
26.4. Operation Flowcharts.....	508
26.5. Data Bus	511
26.6. DMA External Interface.....	514
Chapter 27 Delayed Interrupt.....	517
27.1. Overview	517
27.2. Features.....	517
27.3. Configuration	517
27.4. Register	518
27.5. Operation	518
27.6. Setting.....	519
27.7. Q & A	519
27.8. Caution	519
Chapter 28 Bit Search	521
28.1. Overview	521
28.2. Features.....	521
28.3. Configuration	522
28.4. Register	523
28.5. Operation	525
28.6. Setting.....	527
28.7. Q & A	528
28.8. Caution	529
Chapter 29 MPU / EDSU	531
29.1. Overview	531
29.2. Features.....	533
29.3. Registers.....	534
29.4. Break Functions	556
Chapter 30 I/O Ports	565
30.1. I/O Port Functions MB91V460A.....	565
30.2. I/O Port Functions MB91FV460B	590
30.3. I/O Circuit Types	591
30.4. Port Register Settings.....	593
Chapter 31 External Bus	655
31.1. Overview of the External Bus Interface	655
31.2. External Bus Interface Registers	659
31.3. Setting Example of the Chip Select Area.....	686
31.4. Endian and Bus Access.....	687
31.5. Operation of the Ordinary bus interface.....	704
31.6. Burst Access Operation	716
31.7. Address/data Multiplex Interface	718
31.8. Prefetch Operation.....	721

MB91460 Series

31.9. SDRAM/FCRAM Interface Operation	724
31.10. DMA Access Operation	733
31.11. Bus Arbitration	747
31.12. Procedure for Setting a Register	748
31.13. Notes on Using the External Bus Interface	749
Chapter 32 USART (LIN / FIFO)	751
32.1. Overview	751
32.2. USART Configuration	754
32.3. USART Pins	758
32.4. USART Registers	759
32.5. USART Interrupts	776
32.6. USART Baud Rates	781
32.7. USART Operation	787
32.8. Notes on using USART	803
Chapter 33 I2C Controller	807
33.1. Overview	807
33.2. I2C Interface Registers	809
33.3. I2C Interface Operation	825
33.4. Programming Flow Charts	827
Chapter 34 CAN Controller	831
34.1. Overview	831
34.2. Register Description	833
34.3. Functional Description	862
34.4. CAN Application	866
Chapter 35 Free-Run Timer (FRT)	875
35.1. Overview	875
35.2. Features	875
35.3. Configuration Diagram	876
35.4. Registers	878
35.5. Addresses and Connections	881
35.6. Operation	882
35.7. Settings	884
35.8. Q & A	885
35.9. Caution	888
Chapter 36 Input Capture Unit (ICU)	889
36.1. Overview	889
36.2. Features	889
36.3. Configuration	890
36.4. Register	892
36.5. Operation	896
36.6. Settings	898

36.7. Q&A	899
36.8. Caution	902
Chapter 37 Output Compare Unit (OCU).....	903
37.1. Overview	903
37.2. Features.....	903
37.3. Configuration Diagram.....	904
37.4. Registers.....	905
37.5. Operation	910
37.6. Settings	912
37.7. Q & A	913
37.8. Caution	919
Chapter 38 Reload Timer (RLT)	921
38.1. Overview	921
38.2. Features.....	921
38.3. Configuration	923
38.4. Registers.....	925
38.5. Operation	929
38.6. Setting.....	934
38.7. Q & A	936
38.8. Caution	940
Chapter 39 Programmable Pulse Generator (PPG)	941
39.1. Overview	941
39.2. Features.....	941
39.3. Configuration	944
39.4. Registers.....	946
39.5. Operation	957
39.6. Setting.....	961
39.7. Q & A	963
39.8. Caution	974
Chapter 40 Pulse Frequency Modulator (PFM)	975
40.1. PFM Overview	975
40.2. Reload Counter Registers	978
40.3. Reload Counter Operation	982
40.4. PFM Operation and Setting	985
Chapter 41 Up/Down Counter (UDC).....	987
41.1. Overview	987
41.2. Feature	987
41.3. Configuration	988
41.4. Register	992
41.5. Operation	1000
41.6. Setting.....	1007

MB91460 Series

41.7. Q&A	1009
41.8. Caution	1015
Chapter 42 Sound Generator (SG)	1017
42.1. Overview	1017
42.2. Detailed Structure of the Sound generator	1018
42.3. Registers	1019
Chapter 43 Stepper Motor Controller (SMC)	1025
43.1. Overview	1025
43.2. Registers	1026
43.3. Operation	1034
43.4. Caution	1036
Chapter 44 A/D Converter (ADC) / Range Comparator	1037
44.1. Overview of A/D Converter and A/D Range Comparator	1037
44.2. A/D Converter Block Diagram	1039
44.3. Registers of the A/D Converter and Range Comparator	1040
44.4. Analog Input Connections	1054
44.5. Accuracy and Parameter Setting of the A/D Converter	1055
44.6. Operation Modes of A/D Converter	1058
44.7. ADC Interrupt Generation and DMA Access	1065
44.8. Protection of the ADC Results	1066
44.9. Setting	1068
44.10. Q & A	1070
44.11. Caution	1076
44.12. Range Comparator Overview	1077
44.13. Range Comparator Registers	1078
44.14. Range Comparator Operation	1083
44.15. Range Comparator Interrupt	1086
Chapter 45 D/A Converter (DAC)	1089
45.1. Overview	1089
45.2. Features	1089
45.3. Configuration	1090
45.4. Registers	1091
45.5. Operation	1092
45.6. Setting	1093
45.7. Q & A	1094
45.8. Caution	1095
Chapter 46 Alarm Comparator	1097
46.1. Overview	1097
46.2. Block Diagram	1097
46.3. Alarm Comparator Control/Status Register (ACSR)	1098
46.4. Operation Modes	1099

Chapter 47 LCD Controller.....	1101
47.1. Overview.....	1101
47.2. Features.....	1101
47.3. Configuration	1102
47.4. Registers.....	1104
47.5. Operation	1110
47.6. Setting.....	1116
47.7. Q&A	1117
47.8. Caution	1122
 Chapter 48 Clock Monitor	 1123
48.1. Overview.....	1123
48.2. Features.....	1123
48.3. Configuration	1124
48.4. Register	1125
48.5. Operation	1127
48.6. Settings.....	1128
48.7. Q&A	1128
48.8. Caution	1129
 Chapter 49 Real-Time Clock (RTC)	 1131
49.1. Overview.....	1131
49.2. Features.....	1131
49.3. Configuration	1132
49.4. Registers.....	1133
49.5. Operation	1138
49.6. Setting.....	1140
49.7. Q&A	1141
49.8. Caution	1143
 Chapter 50 Clock Calibration Unit.....	 1145
50.1. Overview.....	1145
50.2. Timing	1146
50.3. Clocks	1147
50.4. Register Description	1148
50.5. Application Note.....	1154
 Chapter 51 Low Voltage Reset/Interrupt (Supply Supervisor)	 1157
51.1. Overview.....	1157
51.2. Features.....	1157
51.3. Registers.....	1158
 Chapter 52 Regulator Control.....	 1161
52.1. Overview.....	1161
52.2. Features.....	1161

MB91460 Series

52.3. Registers.....	1162
Chapter 53 Fixed Mode-Reset Vector / Boot-ROM.....	1165
53.1. Overview.....	1165
53.2. Check for Boot Conditions	1165
53.3. Registers modified by Boot ROM	1170
53.4. Flash Access Mode Switching	1171
53.5. Bootloader Update Strategy	1171
53.6. Chip-ID.....	1173
Chapter 54 Flash Memory	1175
54.1. Overview	1175
54.2. Features.....	1175
54.3. Configuration	1176
54.4. Registers.....	1180
54.5. Access Modes	1180
54.6. Flash Access Mode Switching	1183
54.7. Auto Program Algorithms.....	1185
54.8. Detailed Explanation of Writing to and Erasing Flash Memory.....	1196
54.9. Caution and Notes	1202
Chapter 55 Flash Security.....	1205
55.1. Overview	1205
55.2. Features.....	1205
55.3. Flash Security Vectors.....	1206
55.4. Register	1209
Chapter 56 Embedded Data Flash.....	1213
56.1. Data Flash Features	1213
56.2. Data Flash Block Diagram	1213
56.3. Data Flash Operation Modes.....	1214
56.4. Data Flash access in CPU mode	1215
56.5. Data Flash Registers	1218
56.6. Data Flash Interrupts and DMA Access.....	1223
56.7. Data Flash parallel programming mode.....	1224
56.8. Data Flash Security	1227
Chapter 57 APIX® Controller	1231
57.1. Overview	1231
57.2. Automotive Remote Handler.....	1232
57.3. APIX ® PHY Configuration.....	1244
57.4. DMA transfer request.....	1246
57.5. Automotive Interconnect Pins	1247
57.6. USECASES	1248

Chapter 58 E-Ray	1251
58.1. About this Document	1251
58.2. Overview	1253
58.3. Programmer's Model	1256
58.4. Functional Description	1339
58.5. Appendix	1392
Chapter 59 MediaLB Clock Generation/Bus Interface.....	1407
59.1. MediaLB clock generation block	1407
59.2. PLL interface.....	1409
59.3. Control circuit	1422
59.4. MLBCLK loss detection circuit	1425
59.5. Interrupt control circuit	1429
59.6. Bus interface	1433
Chapter 60 MediaLB (Media Local Bus interface).....	1435
60.1. Overview of the MediaLB.....	1435
60.2. Block Diagram	1437
60.3. MediaLB Registers	1440
60.4. Description of MediaLB Behavior	1490
60.5. Interrupts by MediaLB	1514
60.6. Examples of operating procedures	1517
Chapter 61 I²S (Inter-Integrated Circuit Sound).....	1531
61.1. Overview of the I ² S	1531
61.2. I ² S Registers.....	1531
61.3. Operations of the I ² S	1549
61.4. Interrupts of the I ² S.....	1553
61.5. I ² S Buffer	1554
Chapter 62 USB Mini-Host	1557
62.1. USB Mini-host features.....	1557
62.2. Differences in USB Mini-host.....	1558
62.3. USB Mini-host Block Diagram	1559
62.4. USB Mini-host Registers.....	1560
62.5. USB Mini-host operation	1577
62.6. Token Flow Chart	1588
Chapter 63 USB Function	1591
63.1. USB Function Overview.....	1591
63.2. USB Function Block Diagram	1592
63.3. USB Function Register Description	1593
63.4. Description of the USB function operation	1618
Chapter 64 ShutDown Control.....	1631

MB91460 Series

64.1. Overview	1631
64.2. Standby RAM	1631
64.3. Shutdown Registers	1632
64.4. Shutdown Operation	1638
Chapter 65 Electrical Specification	1647
Appendix A Instruction Lists	1649
A.1 Meaning of Symbols	1650
A.2 Instruction Lists	1657
A.3 Instruction Maps	1667
A.4 Instruction Maps of Instruction Format TYPE-E	1668
Appendix B List of Figures	1671
Appendix C List of Tables	1683
Appendix D Revision History	1691
D.1 Changes in This Revision	1691
D.2 Changes in Preceding Revisions	1691

Chapter 1 Introduction, Handling and Precautions

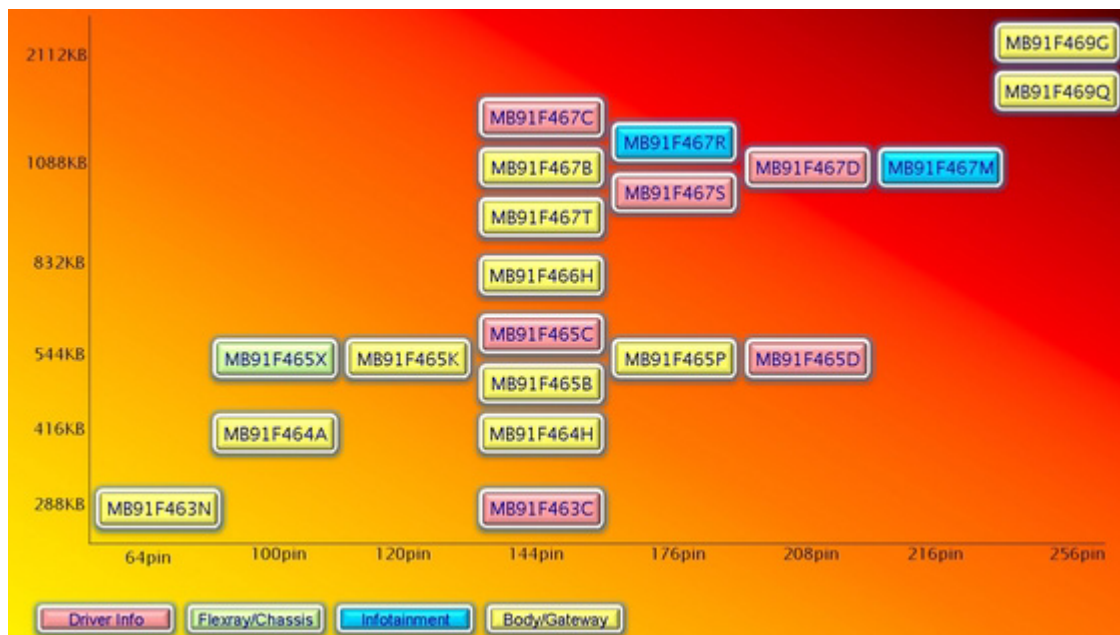
1.1. Introduction

Modern automotive and industrial designs demand higher computing power combined with a high degree of integration. Efficient data collection, data processing and distribution are essential and need a large number of communication interfaces such as CAN, FlexRay and LIN as well as a large on-chip memory capacity.

Fujitsu's latest MB91460 family has been designed in close co-operation with major automotive customers worldwide and inherits the high-performance core of Fujitsu's proprietary FR¹ MCU architecture. This high computing performance combined with powerful peripheral functions and features supports a wide range of applications and offers a high grade of flexibility to provide superior solutions for automotive, computer peripheral, industrial, and consumer applications.

Some devices offer an external bus interface which can be connected to Fujitsu's stand alone Flexray controller or to the latest generation of graphics controllers in order to build full-featured dashboards, driver information and advanced driver assistance systems.

Figure 1.1-1 MB91460 series roadmap



1. FR, the abbreviation of FUJITSU RISC controller, is a line of products of Fujitsu Semiconductor Limited.

1.2. Handling Precautions

1.2.1 Precautions for Product Design

This section describes precautions when designing electronic equipment using semiconductor devices.

■ Absolute Maximum Ratings

Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of certain established limits, called absolute maximum ratings. Do not exceed these ratings.

■ Recommended Operating Conditions

Recommended operating conditions are normal operating ranges for the semiconductor device. All the device's electrical characteristics are warranted when operated within these ranges.

Always use semiconductor devices within the recommended operating conditions. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their sales representative beforehand.

■ Processing and Protection of Pins

These precautions must be followed when handling the pins which connect semiconductor devices to power supply and input/output functions.

1. Preventing Over-Voltage and Over-Current Conditions

Exposure to voltage or current levels in excess of maximum ratings at any pin is likely to cause deterioration within the device, and in extreme cases leads to permanent damage of the device. Try to prevent such overvoltage or over-current conditions at the design stage.

2. Protection of Output Pins

Shorting of output pins to supply pins or other output pins, or connection to large capacitance can cause large current flows. Such conditions if present for extended periods of time can damage the device.

Therefore, avoid this type of connection.

3. Handling of Unused Input Pins

Unconnected input pins with very high impedance levels can adversely affect stability of operation. Such pins should be connected through an appropriate resistance to a power supply pin or ground pin.

If unused input pins are left open, abnormal operation may result. Any unused input pins should be connected to pull-up or pull-down resistor (2 k Ω to 10 k Ω) or enable internal pullup or pulldown resistors (PPER/PPCR) before the input enable (PORTEN) is activated by software. The mode pins MD_x can be connected to VSS5 or VDD5 directly. Unused ALARM input pins can be connected to AVSS5 directly.

■ Latch-up

Semiconductor devices are constructed by the formation of P-type and N-type areas on a substrate. When subjected to abnormally high voltages, internal parasitic PNP junctions (called thyristor structures) may be formed, causing large current levels in excess of several hundred mA to flow continuously at the power supply pin. This condition is called latch-up.

CAUTION: The occurrence of latch-up not only causes loss of reliability in the semiconductor device, but can cause injury or damage from high heat, smoke or flame. To prevent this from happening, do the following:

1. Be sure that voltages applied to pins do not exceed the absolute maximum ratings. This should include attention to abnormal noise, surge levels, etc.
2. Be sure that abnormal current flows do not occur during the power-on sequence.

■ Power supply pins

In MB91460 series, devices including multiple power supply pins and ground pins are designed as follows;

MB91460 Series

pins necessary to be at the same potential are interconnected internally to prevent malfunctions such as latchup.

All of the power supply pins and ground pins must be externally connected to the power supply and ground respectively in order to reduce unnecessary radiation, to prevent strobe signal malfunctions due to the ground level rising and to follow the total output current ratings. Furthermore, the power supply pins and ground pins of the MB91460 series must be connected to the current supply source via a low impedance.

It is also recommended to connect a ceramic capacitor of approximately 0.1 μF as a bypass capacitor between power supply pin and ground pin near this device.

This series has a built-in step-down regulator. Connect a bypass capacitor of 4.7 μF (use a X7R ceramic capacitor) to VCC18C pin for the regulator.

■ Crystal oscillator circuit

Noise in proximity to the X0 (X0A) and X1 (X1A) pins can cause the device to operate abnormally. Printed circuit boards should be designed so that the X0 (X0A) and X1 (X1A) pins, and crystal oscillator, as well as bypass capacitors connected to ground, are located near the device and ground.

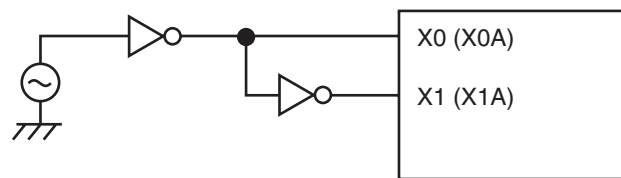
It is recommended that the printed circuit board layout be designed such that the X0 and X1 pins or X0A and X1A pins are surrounded by ground plane for the stable operation.

Please request the oscillator manufacturer to evaluate the oscillational characteristics of the crystal and this device.

■ Notes on using external clock

When using the external clock, it is necessary to simultaneously supply the X0 (X0A) and the X1 (X1A) pins. In the described combination, X1 (X1A) should be supplied with a clock signal which has the opposite phase to the X0 (X0A) pins. At X0 and X1, a frequency up to 16 MHz is possible.

Figure 1.2-1 Opposite phase external clock driving



■ Mode pins (MD_x)

These pins should be connected directly to the power supply or ground pins. To prevent the device from entering test mode accidentally due to noise, minimize the lengths of the patterns between each mode pin and power supply pin or ground pin on the printed circuit board as possible and connect them with low impedance.

■ Observance of Safety Regulations and Standards

Most countries in the world have established standards and regulations regarding safety, protection from electromagnetic interference, etc. Customers are requested to observe applicable regulations and standards in the design of products.

■ Fail-Safe Design

Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

■ Precautions Related to Usage of Devices

FUJITSU SEMICONDUCTOR semiconductor devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, personal or household devices, etc.).

CAUTION: Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

■ Notes on operating in PLL clock mode

If the oscillator is disconnected or the clock input stops when the PLL clock is selected, the microcontroller may continue to operate at the free-running frequency of the self-oscillating circuit of the PLL. However, this self-running operation cannot be guaranteed.

■ Pull-up control

The AC standard is not guaranteed in case a pull-up resistor is connected to the pin serving as an external bus pin.

1.2.2 Precautions for Package Mounting

Package mounting may be either lead insertion type or surface mount type. In either case, for heat resistance during soldering, you should only mount under FUJITSU SEMICONDUCTOR's recommended conditions. For detailed information about mount conditions, contact your sales representative.

■ Lead Insertion Type

Mounting of lead insertion type packages onto printed circuit boards may be done by two methods: direct soldering on the board, or mounting by using a socket.

Direct mounting onto boards normally involves processes for inserting leads into through-holes on the board and using the flow soldering (wave soldering) method of applying liquid solder. In this case, the soldering process usually causes leads to be subjected to thermal stress in excess of the absolute ratings for storage temperature. Mounting processes should conform to FUJITSU SEMICONDUCTOR recommended mounting conditions.

If socket mounting is used, differences in surface treatment of the socket contacts and IC lead surfaces can lead to contact deterioration after long periods. For this reason it is recommended that the surface treatment of socket contacts and IC leads be verified before mounting.

■ Surface Mount Type

Surface mount packaging has longer and thinner leads than lead-insertion packaging, and therefore leads are more easily deformed or bent. The use of packages with higher pin counts and narrower pin pitch results in increased susceptibility to open connections caused by deformed pins, or shorting due to solder bridges.

You must use appropriate mounting techniques. FUJITSU SEMICONDUCTOR recommends the solder reflow method, and has established a ranking of mounting conditions for each product. Users are advised to mount packages in accordance with FUJITSU SEMICONDUCTOR ranking of recommended conditions.

■ Lead-Free Packaging

CAUTION: When ball grid array (BGA) packages with Sn-Ag-Cu balls are mounted using Sn-Pb eutectic soldering, junction strength may be reduced under some conditions of use.

■ Storage of Semiconductor Devices

Because plastic chip packages are formed from plastic resins, exposure to natural environmental conditions

will cause absorption of moisture. During mounting, the application of heat to a package that has absorbed moisture can cause surfaces to peel, reducing moisture resistance and causing packages to crack. To prevent, do the following:

1. Avoid exposure to rapid temperature changes, which cause moisture to condense inside the product. Store products in locations where temperature changes are slight.
2. Use dry boxes for product storage. Products should be stored below 70% relative humidity, and at temperatures between 5 °C and 30 °C.
When you open Dry Package that recommends humidity 40% to 70% relative humidity.
3. When necessary, FUJITSU SEMICONDUCTOR packages semiconductor devices in highly moisture-resistant aluminum laminate bags, with a silica gel desiccant. Devices should be sealed in their aluminum laminate bags for storage.
4. Avoid storing packages where they are exposed to corrosive gases or high levels of dust.

■ Baking

Packages that have absorbed moisture may be de-moisturized by baking (heat drying). Follow the FUJITSU SEMICONDUCTOR recommended conditions for baking.

Condition: 125 °C/24 h

■ Static Electricity

Because semiconductor devices are particularly susceptible to damage by static electricity, you must take the following precautions:

1. Maintain relative humidity in the working environment between 40% and 70%.
Use of an apparatus for ion generation may be needed to remove electricity.
2. Electrically ground all conveyors, solder vessels, soldering irons and peripheral equipment.
3. Eliminate static body electricity by the use of rings or bracelets connected to ground through high resistance (on the level of 1 MΩ).
Wearing of conductive clothing and shoes, use of conductive floor mats and other measures to minimize shock loads is recommended.
4. Ground all fixtures and instruments, or protect with anti-static measures.
5. Avoid the use of styrofoam or other highly static-prone materials for storage of completed board assemblies.

1.2.3 Precautions for Use Environment

Reliability of semiconductor devices depends on ambient temperature and other conditions as described above. For reliable performance, do the following:

1. Humidity
Prolonged use in high humidity can lead to leakage in devices as well as printed circuit boards. If high humidity levels are anticipated, consider anti-humidity processing.
2. Discharge of Static Electricity
When high-voltage charges exist close to semiconductor devices, discharges can cause abnormal operation. In such cases, use anti-static measures or processing to prevent discharges.
3. Corrosive Gases, Dust, or Oil
Exposure to corrosive gases or contact with dust or oil may lead to chemical reactions that will adversely affect the device. If you use devices in such conditions, consider ways to prevent such exposure or to protect the devices.
4. Radiation, Including Cosmic Radiation
Most devices are not designed for environments involving exposure to radiation or cosmic radiation. Users should provide shielding as appropriate.
5. Smoke, Flame

Plastic molded devices are flammable, and therefore should not be used near combustible substances. If devices begin to smoke or burn, there is danger of the release of toxic gases.

Customers considering the use of FUJITSU SEMICONDUCTOR products in other special environmental conditions should consult with sales representatives.

Please check the latest handling precautions at the following URL.

<http://edevise.fujitsu.com/fj/handling-e.pdf>

1.3. Software Precautions

1.3.1 External Reset

By inputting “L” to INITX, ensure the clock oscillation stabilization time. Keep INITX low for at least 10 ms during power-on or if the main oscillator was stopped before.

1.3.2 Switching of multi-purpose port

Use PFR (Port function register) and EPFR (Extra Port Function Register) to switch between General Purpose In/Out (GPIO) and resource operation.

1.3.3 Low-power-consumption mode

For standby mode, enable synchronous standby (TBCR.SYNCS=“1”) and then use the following sequence:

```
LDI    #value_of_STCR,r0    ; value of Standby Control register
LDI    #_STCR,r12           ; address of Standby Control register (481H)
STB    r0,@r12              ; write to STCR
LDUB   @r12,r0              ; STCR read for synchronous standby
LDUB   @r12,r0              ; dummy read STCR again
NOP                    ; NOP x 5 for pipeline clear
NOP
NOP
NOP
NOP
```

If monitor debugger is used, the following should be avoided.

- Do not set breakpoints in the command sequence above.
- Do not conduct stepwise execution in the command sequence above.

1.3.4 Watchdog timer function

For more information about the software and the hardware watchdogs, see “[Chapter 20 Software Watchdog Timer \(Page No.389\)](#)” and “[Chapter 21 Hardware Watchdog Timer \(Page No.399\)](#)”.

1.3.5 Register against read-modify-write command

Read-Modify-Write (R-M-W) instructions are used to manipulate up to 4 bits within a register.

BANDL	#u4,@Ri	;	Modify the lower 4 bit with AND function
BANDH	#u4,@Ri	;	Modify the upper 4 bit with AND function
BORL	#u4,@Ri	;	Modify the lower 4 bit with OR function
BORH	#u4,@Ri	;	Modify the upper 4 bit with OR function
BEORL	#u4,@Ri	;	Modify the lower 4 bit with EOR function
BEORH	#u4,@Ri	;	Modify the upper 4 bit with EOR function
BTSTL	#u4,@Ri	;	Test lower 4 bits (no reg.write)
BTSTH	#u4,@Ri	;	Test upper 4 bits (no reg.write)

#u4: unsigned 4-bit value (0x0 - 0xF) Ri: CPU register R1...R15

The CPU reads the register with 8-bit access, changes the 4 bits and writes back the modified 8-bit value.

The hardware decodes the read access of R-M-W especially regarding read operation to flags. For example, to avoid accidentally clearing of interrupt or status flags, those flags always read “1” during R-M-W. The following write “1” to the flag has no effect, because status or interrupt flags cannot be set by user software.

The following table lists registers, which should **not** be accessed using Read-Modify-Write. To write those registers, use normal byte / half-word / word access.

Table 1.3-1 Registers, which should not be accessed with Read-Modify-Write

Module	Register	Read-Modify-Write Item
USART	ECCRnn ¹	If SMRnn_MD == 2 (synchronous mode), R-M-W can be used. If SMRnn_MD != 2, the bit ECCRnn_SSM is masked, R-M-W reads “0” and writes back “0”. So SSM can be cleared accidentally.
USART	RDRnn TDRnn ¹	RSR and TDR are separate registers under the same address. RDR is read-only, TDR is write only. R-M-W would copy bits from RDR to TDR.
A/D Converter 0 A/D Converter 1	AD0CS0 AD1CS0	On some new devices ² , bit ADCS[0] has different meanings at read or write: Read ADCS[0] returns ACH0 (active channel number, bit 0). Write ADCS[0] modifies ACHMD (active channel register mode). R-M-W can clear or set ACHMD by accident.

1. nn = 00 to 15, number of USART macro
2. MB91FV460B, MB91F467E, MB91F467P, MB91F469Q

Note: In the preceeding revisions of this hardware manual, the SMRnn register in the USARTs were prohibited for R-M-W access. This information was wrong, SMRnn can be accessed with R-M-W.

1.3.6 Caution: Writing to registers which include a status flag

Writing to a register including a status flag (in particular, interrupt flag), note that the status flag should not be cleared unintentionally. Especially writing a “0” to a status flag may clear it by accident. Writing a “1” has no effect, because the flag can not be set by the user.

By using bit commands (Read-Modify-Write), the interrupt flags are always read as “1”.

The following table shows examples of registers which mostly include both of control bits and status flags.

Table 1.3-2 Registers, which include status flags (examples)

Module	Register	Status flags
Timebase timer	TBCR	TBIF Interrupt flag
PLL control	PLLCTRL	GRDN, GRUP Gear-down, gear-up interrupt flags
Clock Supervisor	CSVCR	MM, SM Main clock missing, Sub clock missing flags ¹
Hardware Watchdog	HWWD	CPUF hardware watchdog timeout flag
Main osci. stabi timer	OSCR	WIF Interrupt flag
Sub osci. stabi timer	WPCR	WIF Interrupt flag
External Interrupts	EIRR0,1,2,3	ER0-ER7 External interrupt request flags
USART	SSRnn ²	TDRE, RDRF, FRE, ORE, PE status and error flags
USART	ESCRnn ²	LBD status flag
USART	ECCRnn ²	RBI, TBI - but see 1.3.5 Register against read-modify-write command (Page No.7) above!

Module	Register	Status flags
USART	SSRnn ^{2,3}	ETINT End of transmission interrupt
I2C	IBCRn ⁴	BER, INT bus error flag, interrupt flag
Reload Timer	TMCSRnn ⁵	UF interrupt flag

1. MM, SM are writeable only on MB91FV460B, MB91F467E
2. nn = 00 to 15, number of USART macro
3. New devices only: MB91FV460B, MB91F467E, MB91F467P
4. n = 0 to 7, number of I2C macro
5. nn = 00 to 15, number of Reload Timer macro

Note: There are much more control- and status registers.

1.3.7 Caution: Delayed writing to registers which include a status flag

Writing to a register including a status flag (in particular, interrupt flag) in order to control the function, note that the actual writing to the registers may be delayed. This is because of using write buffers on the busses to the resources which accept a write access from CPU immediately but can access the resource registers delayed.

In this case it can happen that (within an Interrupt Service Routine) the interrupt flag is cleared by writing “0” to the register and the ISR is completed with RETI, but the interrupt flag is still active and the ISR is executed again.

To synchronize the access to the resources on this architecture, please follow this recommendation:

Use a read access (byte or halfword) to the RBSYNC address to synchronize the CPU operation (e.g. the interrupt acceptance of the CPU) to a preceding write access to the resources on R-bus (e.g. to an interrupt flag) on following addresses (0x0000-0x01FF, 0x0280-0x038B, 0x0394-0x03B7, 0x0400-0x063F, 0x0688-0x075F and 0x0C00-0x0FFF).

Use a read access (byte or halfword) to the CBSYNC address to synchronize the CPU operation (e.g. the interrupt acceptance of the CPU) to a preceding write access to the CANs on D-bus (e.g. to an interrupt flag) on following addresses (0xC000-0xFFFF).

1.3.8 Caution: Miscellaneous notes

The following miscellaneous items should be noted:

- Always write “1” to bit IOS[1] of IOS register (address 0x0C03) .
- Port multiplexing registers PPMUX and PPMUX2 (available on MB91FV460B and MB91460P- and T-Series) can be written only once after reset.

1.4. Notes on Debugger

1.4.1 Execution of the RETI Command

If single-step execution is used in an environment where an interrupt occurs frequently, the corresponding interrupt handling routine will be executed repeatedly to the exclusion of other processing. This will prevent the main routine and the handlers for low priority level interrupts from being executed (For example, if the time-base timer interrupt is enabled, stepping over the RETI instruction will always break on the first line of the time-base timer interrupt handler).

Disable the corresponding interrupts when the corresponding interrupt handling routine no longer needs debugging.

1.4.2 Break function

If the range of addresses that cause a hardware break (including event breaks) is set to the address of the current system stack pointer or to an area that contains the stack pointer, execution will break after each instruction regardless of whether the user program actually contains data access instructions.

To prevent this, do not set (word) access to the area containing the address of the system stack pointer as the target of the hardware break (including an event breaks).

1.4.3 Operand break

It may cause malfunctions if a stack pointer exists in the area which is set as the DSU operand break. Do not set the access to the areas containing the address of system stack pointer as a target of data event break.

1.4.4 Caution: PS register

As the PS register is processed in advance by some instructions, when the debugger is being used, the exception handling may result in execution breaking in an interrupt handling routine or the displayed values of the flags in the PS register being updated.

As the microcontroller is designed to carry out reprocessing correctly upon returning from such an EIT event, the operation before and after the EIT always proceeds according to specification.

- **The following behavior may occur if any of the following occurs in the instruction immediately after a DIV0U/DIV0S instruction:**

- (a) a user interrupt or NMI is accepted;
- (b) single-step execution is performed;
- (c) execution breaks due to a data event or from the emulator menu.
 1. D0 and D1 flags are updated in advance.
 2. An EIT handling routine (user interrupt/NMI or emulator) is executed.
 3. Upon returning from the EIT, the DIV0U/DIV0S instruction is executed and the D0 and D1 flags are updated to the same values as those in 1.

- **The following behavior occurs when an ORCCR, STILM, MOV Ri,PS instruction is executed to enable a user interrupt or NMI source while that interrupt is in the active state.**

1. The PS register is updated in advance.
2. An EIT handling routine (user interrupt/NMI or emulator) is executed.
3. Upon returning from the EIT, the above instructions are executed and the PS register is updated to the same value as in 1.

1.5. How to Use This Document

1.5.1 Main terminology

This table shows main terminology used for FR-family CPU hardware manual.

Table 1.5-1 Main terminology

Term	Meaning
I-bus	32-bit-wide bus for internal instruction. Since FR-family CPU series employ internal Harvard architecture, instruction and data are independent busses. For I-bus, Harvard/Princeton-bus-converter is connected. Also connected to I-bus is EDSU/MPU.
D-bus	Internal 32-bit-wide data bus. For D-bus, bit search module, Harvard/Princeton-bus-converter, R-bus interface (32-bit \leftrightarrow 16-bit Bus-converter), CAN modules and EDSU/MPU are connected.
F-bus	Internal 32-bit-wide bus. F-bus is connected to embedded Flash/ROM and embedded RAM.
R-bus	Internal 16-bit-wide data bus. R-bus is connected to D-bus via R-bus-converter. For R-bus, peripheral function, I/O, clock generator and interrupt controller are connected.
X-bus	32-bit-wide address and data bus. Via bus-converter for external bus, it accesses to external bus.
OSCMAN	Output from Main Oscillator (typically 4MHz), connected to the main clock supervisor.
OSCSUB	Output from Sub Oscillator (typically 32 kHz), connected to the sub clock supervisor.
CLKRC 100kHz	This is a clock which is connected to the RC Oscillator (Typical value 100kHz). This clock is connected to the RC based watchdog and can be source for real time clock.
CLKRC 2MHz	Output from RC Oscillator 2MHz
CLKRC	RC Clock, either 100kHz (CLKRC 100kHz) or 2MHz (CLKRC 2MHz)
CLKMAIN (Main Clock)	This is a clock which acts as a benchmark for LSI operation triggered by 4 MHz main oscillator. This clock is output of main clock supervisor and connected to main clock oscillation stabilization timer. It can be source for base clock and real time clock.
CLKSUB (Sub clock)	This is a clock which acts as a benchmark for LSI operation triggered by 32 kHz sub oscillator. This clock is output of sub clock supervisor and is connected to sub oscillation stabilization timer. It can be source for real-time clock and for base clock.
CLKSUBRC	Sub clock for Base Clock generation. This clock is either CLKSUB or CLKRC.
CLKVCO	Output from PLL Oscillator
CLKPLLFB	PLL feedback
CLKPLL	Output from the PLL interface
CLKMOD	Output from Clock Modulator (modulated CLKPLL)
CLKMAINPLL	Clock output after clock modulator (either modulated CLKPLL or unmodulated CLKPLL)
Φ (Base clock)	Base clock is the maximum speed basis clock which generates CLKB, CLKP and CKLT in the clock generator. Clock source can be CLKMAIN divided by 2, CLKMAINPLL or CLKSUBRC.
CLKB (CPU clock)	CPU clock is the clock which is referred by CPU, embedded ROM, embedded RAM, bit search module and internal bus (I-bus, D-bus, F-bus and X-bus) operations. Generated from the base clock Φ in the clock generator.
CLKP (Peripheral clk.)	Peripheral clock is the clock which is referred by each peripheral function (peripheral functions other than bit search module and CAN) connected to R-bus and R-bus, clock control, interrupt controller, I/O port and external interrupt input operations. Generated from the base clock Φ in the clock generator.

Term	Meaning
CLKT (External bus clock)	External bus clock is the clock which is referred by external expansion bus interface connected to X-BUS and external clock output operations. Generated from the base clock Φ in the clock generator.
CLKCAN (CAN clock)	CAN clock is the clock which is referred by the CAN modules. Generated from the non modulated PLL output clock to ensure operation within CAN network oscillation tolerances.
Main clock mode	Mode which runs based on Main clock. This Main clock mode has status such as Main-RUN, Main-SLEEP, Main-STOP, oscillation stabilization wait RUN, oscillation stabilization wait reset and program reset.
Sub clock mode	Mode which runs based on Sub clock. This Sub clock mode has status such as Sub-RUN, Sub-SLEEP, Sub-STOP, Sub clock oscillation stabilization wait RUN and program reset.
Main-RUN	Main-RUN is the state which is in Main clock mode and also all circuits are operable.
Main-SLEEP	Main-SLEEP is the state which is in Main clock mode and just the peripherals are operable.
Main-STOP	Main-STOP is the state which is in Main clock mode and all circuits are stopped.
Sub-RUN	Sub-RUN is the state which is in Sub clock mode and also all circuits are operable.
Sub-SLEEP	Sub-SLEEP is the state which is in Sub clock mode and just the peripherals are operable.
Sub-STOP	Sub-STOP is the state which is in Sub clock mode and all circuits are stopped.
RTC-Mode	Real Time Clock mode is a kind of STOP state with the Real Time Clock and its clock source operating and all other circuits stopped.
ShutDown State	ShutDown is a kind of STOP state with power supply for most of the circuits and memories switched off. Real Time Clock and its clock source as well as the RC watchdog can operate. Available on MB91F467E only.
Oscillation stabilization time	Upon the reset (INITX, RST), return from STOP, return from PLL abnormal operation, generation of watchdog and during Main clock stop, it takes oscillation stabilization time for Main clock. Time base timer counts the time.
Main clock oscillation stabilization wait	Wait time until Main clock oscillates after Main clock stops in Sub clock mode. Main clock oscillation stabilization timer counts the time.
D-RAM	Data RAM
ID-RAM	Instruction-Data RAM
I-Cache	Instruction-Cache on external bus interface
F-Cache	Flash-Cache, Instruction-Cache on flash memory interface
DMAC	DMA Controller
FRT	Free-Run Timer
VDD	Internal core supply.
VSS	Digital supply ground voltage.
SWWD	Software Watchdog
HWWD	Hardware Watchdog
PLL, PLL2	Main PLL, Flexray PLL
Main Clock SV	Main Clock Supervisor
Sub Clock SV	Sub Clock Supervisor
IA	Instruction address
OA	Operand address
DT	Data

1.5.2 Access size and address position

Address	Address offset value/Register name				Block
	+0	+1	+2	+3	
0000B0 _H	RCR1 [W] B,H,W 00000000	RCR0 [W] B,H,W 00000000	UDCR1 [R] B,H,W 00000000	UDCR0 [R] B,H,W 00000000	Up/down counter 0,1
0000B4 _H	CCR0 [R/W] B,H,W 00000000	CCR1 [R/W] B,H,W 00001000	-	CSR0 [R/W] B,H,W 00000000	
0000B8 _H	CCR0 [R/W] B,H,W 00000000	CCR1 [R/W] B,H,W 00001000	-	CSR1 [R/W] B,H,W 00000000	

Byte access, Half-word access, and Word access are allowed.

There are three kinds of accesses such as Byte access, Half-word access and Word access. However, note that some registers have restricted access. For more information, see “3.2. I/O Map (Page No.36)” or “Detail Description of Register” in each chapter.

B,H,W : Byte access, Half-word access, and Word access are allowed.
 B : Byte access (Be sure to access by Byte.)
 H : Half-word access (Be sure to access by Half-word.)
 W : Word access (Be sure to access by Word.)
 B, H : Byte access, Half-word access only (Word access is not allowed.)
 H, W : Half-word access, Word access only (Byte access is not allowed.)

Reference

The following describes address position to access.

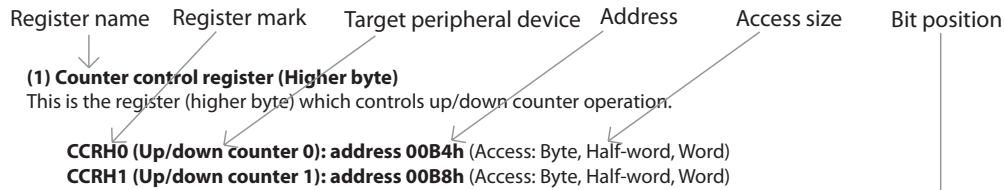
- In Word access, address becomes multiple of 4. (Lowest order 2 bits mandatorily become “00”.)
- In Half-word access, address becomes multiple of 2. (Lowest order 1 bit mandatorily becomes “0”.)
- In Byte access, address will not be changed.

Therefore, for example, make RCR0 register to use Half-word access,

For address 0B0H, RCR1+RCR0 register is accessed.

(When address offset is +1 and +2, (Example: RCR0+UDCR1) Half-word access is not allowed.)

1.5.3 About access size and bit position



15	14	13	12	11	10	9	8	bit
M16E/Reserved	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	
0	0	0	0	0	0	0	0	Initial value
R/W *	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

bit15: Enable 16-bit mode

M16E (CCRH0 only)	Enable 16-bit mode
0	8-bit x 2-channel mode (8-bit mode)
1	16-bit x 1-channel mode (16-bit mode)

*: CCRH1: Reserved (Always write 0 for writing. The read value is indeterminate).

When access size changes, bit position changes.

- In the case that address offset value is +0 (Example: CCRH0 register)

Access size	Address	Bit position							
Byte	0B4 _H +0 _H	07	06	05	04	03	02	01	00
Half-word	0B4 _H +0 _H	15	14	13	12	11	10	09	08
Word	0B4 _H +0 _H	31	30	29	28	27	26	25	24
Bit name		M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0

- In the case that address offset value is +1 (Example: CCRL0 register)

Access size	Address	Bit position							
Byte	0B4 _H +1 _H	07	06	05	04	03	02	01	00
Half-word	0B4 _H +0 _H	07	06	05	04	03	02	01	00
Word	0B4 _H +0 _H	23	22	21	20	19	18	17	16
Bit name		Reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0

- In the case that address offset value is +2 (Example: UDCR1 register)

Access size	Address	Bit position							
Byte	0B0 _H +2 _H	07	06	05	04	03	02	01	00
Half-word	0B0 _H +2 _H	15	14	13	12	11	10	09	08
Word	0B0 _H +0 _H	15	14	13	12	11	10	09	08
Bit name		D15	D14	D13	D12	D11	D10	D9	D8

- In the case that address offset value is +3 (Example: UDCR 1 register)

Access size	Address	Bit position							
Byte	0B0 _H +3 _H	07	06	05	04	03	02	01	00
Half-word	0B0 _H +2 _H	07	06	05	04	03	02	01	00
Word	0B0 _H +0 _H	07	06	05	04	03	02	01	00
Bit name		D7	D6	D5	D4	D3	D2	D1	D0

1.5.4 Meaning of Bit Attribute Symbols

■ Meaning of Bit Attribute Symbols

R	: Readable
W	: Writable
RM	: Reading operation during read/modify/write operation.
"/" (Slash) R/W	: Readable and writable. (The read value is the value written.)
"," (comma) R,W	: Values are different between read and write. (The read value is different from the value written.)
R0	: The read value is "0".
R1	: The read value is "1".
W0	: Always write "0".
W1	: Always write "1".
(RM0)	: read/modify/write operation reads "0".
(RM1)	: read/modify/write operation reads "1".
RX	: The read value is indeterminate. (Reserved bit or undefined bit)
WX	: Writing does not affect the operation. (Undefined bit)
X	: Don't care
-	: Holds it's value

• Example of how R/W is used

- R/W : Readable and writable. (The read value is the value written.)
- R,W : Readable and writable. (The read value and written value are different.)
- R,RM/W : Readable and writable. (The read value and written value are different. Read/modify/write command reads the value written.) Example: port data register
- R(RM1),W : Readable and writable. (The read value and written value are different. Read/modify/write command reads 1.) Example: interrupt flag
- R/WX : Read-only (Read-only. Writing does not affect the operation.)
- R1,W : Write-only (Write-only. The read value is 1.)
- R0,W : Write-only (Write-only. The read value is 0.)
- RX,W : Write-only (Write-only. The read value is indeterminate.)
- R/W0 : Reserved bit (The written value is 0. The read value is the value written.)
- R0/W0 : Reserved bit (The written value is 0. The read value is 0.)
- R1,W0 : Reserved bit (The written value is 0. The read value is 1.)
- RX,W0 : Reserved bit (The written value is 0. The read value is indeterminate.)
- R/W1 : Reserved bit (The written value is 1. The read value is the value written.)
- R1/W1 : Reserved bit (The written value is 1. The read value is 1.)
- R0,W1 : Reserved bit (The written value is 1. The read value is 0.)
- RX,W1 : Reserved bit (The written value is 1. The read value is indeterminate.)
- RX/WX : Undefined bit (The read value is indeterminate. Writing does not affect the operation.)
- R0/WX : Undefined bit (The read value is 0. Writing does not affect the operation.)

Chapter 2 MB91460 Rev.A/Rev.B Overview

2.1. Overview

There are 2 evaluation devices for MB91460 series: MB91V460A ("Rev.A") and MB91FV460B ("Rev.B").

MB91FV460B has the same features as MB91V460A and adds a lot of additional components in order to support infotainment applications and to evaluate derivatives having various differences in port/resource assignments, clock supply, etc. For this purpose, MB91FV460B can be configured for 7 "Series Modes".

2.2. Features

2.2.1 FR-family CPU Core

- 32-bit RISC, load/store architecture, 5-stage pipeline
- Maximum operating frequency: Core clock = 100MHz (device dependent)
(Source oscillation= 4MHz, multiplied by 25 (PLL clock multiplier method))
- General-purpose registers: 16 x 32 bits
- 16-bit fixed-length instruction (Base instruction)
- 32-bit linear address space: 4 Gbytes
- Instructions suitable for embedded application
- Transfer command between memories
- Bit-processing instruction
- Barrel-shift instructions
- Instructions supporting C-language
- Function's enter command /exit command
- Multi-load/store command of register contents
- Assembler statement is also easily available
Register's interlock function
- Multiplier's embedded application/command level support
- Signed 32-bit multiplication: 5 cycles
- Signed 16-bit multiplication: 3 cycles
- Interrupt (PC/PS are saved): 6 cycles (16 priority level)
- Harvard architecture enables simultaneous execution of program access and data access
- Memory protection function
- Embedded debug support
- Commands compatible with FR family

2.2.2 Instruction Cache

- 2 way set associative I-cache on External Bus
 - Up to 4 kByte integrated
 - 4 words (16 bytes) per set
 - Variable capacity (4/2/1 kB)
 - Lock function enabling programs to be resident
 - Available as instruction RAM requiring no wait state when not used as an instruction cache
- Direct mapped I-cache on the Main Program/Data Flash
 - Up to 16 kByte integrated

- Variable capacity (16/8/4/2/1 kB)
- Lock function enabling programs to be resident

2.2.3 Interrupt Controller

- A total of up to 33 external interrupt lines (1 nonmaskable interrupt pin, 24 normal interrupt pins, 8 interrupt pins shared (with peripheral inputs for Wake Up from STOP state, e.g. CAN RX))
- Interrupts from internal peripherals (256 interrupt vectors)
- Priority levels programmable for normal interrupt lines excluding the nonmaskable one (16 levels)
- Capable of using the normal interrupt and nonmaskable interrupt pins for Wake Up from STOP state

2.2.4 Internal Data RAM

- Up to 64 kBytes integrated
- Zero wait state for read/write access
- Referenced as D-RAM in this manual

2.2.5 Internal Instruction/Data RAM

- Up to 64 kBytes integrated
- Zero wait state for read/write access of instructions
- One wait state for read/write access of data
- Referenced as ID-RAM in this manual

2.2.6 Embedded Instruction/Data Memory

- Up to 2112 KByte (Flash or Mask ROM)
- Programmable wait state for read/write access
- Flash/ROM security

2.2.7 External Bus Interface

- 8 chip select areas with individual area size, data bus width selection (8, 16, 32-bit) and wait
- Address bus up to 32 bit wide
- Programmable auto-wait function or external wait input (RDY)
- Basic bus cycles : 2 cycles
- Prefetch function
- Burst access function

2.2.8 DMA Controller

- Four transfer modes supported: single/block, burst, continuous transfer, and fly-by
- 5 channels (4 channels for external-to-external transfer)
- 3 types of transfer sources (external pins/internal peripherals/and software)
- Up to 128 selectable internal transfer sources
- Addressing mode: Specifying up to 32-bit addresses (Increment/decrement/fixed)
- Transfer mode (Demand transfer/burst transfer/step transfer/block transfer)
- Fly-by transfer supported (between external I/O and memory)
- Transferred data size selectable from among 8, 16, and 32 bits

MB91460 Series

2.2.9 Peripheral Function

- General-purpose ports
 - General purpose, configurable CMOS input/output ports
 - N channel open drain port out of above: up to 16 (for I²C)
- A/D converter : up to 54 channels (2 units)
 - ADC0: 10-bit resolution, up to 32 channels
 - ADC1: 10-bit resolution, up to 22 channels
 - Series-parallel type
 - Minimum conversion time: 1us
 - Single conversion mode
 - Continuous conversion mode
 - Stop conversion mode
 - Activation by software or external trigger can be selected
 - Reload timer 7 and A/D Converter co-operate
- D/A converter : 2 channels
 - R-2R type
 - Resolution: 10 bits
 - Conversion rate: 0.45us (when 20 pF load is applied)
 - Conversion rate: 2us (when 100 pF load is applied)
- Alarm comparator : 2 channels
 - Monitors an external voltage and generates an interrupt in case of a voltage lower or higher than the defined thresholds
 - Status is readable, interrupts can be masked separately
- External interrupt input : 32 channels
 - Can be programmed to be edge sensitive or level sensitive
 - Interrupt mask and request pending bits per channel
 - 6 channels combined with CAN RX for wakeup
- Nonmaskable interrupt (NMI) : 1 channel
 - Highest priority of all user interrupts
- Bit search module (using REALOS)
 - Function to search the first bit position of "1", "0", "Changed" from MSB (most significant bit) within 1 word
- Up/down counter : 16 bits x 2 channels (8 bits x 4 channels)
 - Timer mode, up/down count mode, phase difference mode (x2, x4)
 - Includes clock prescaler ($f_{RES}/2^1$, $f_{RES}/2^3$)

- **Reload timer : 16 bits x 16 channels**
 - 16-bit reload counter
 - Includes clock prescaler ($f_{RES}/2^1$, $f_{RES}/2^3$, $f_{RES}/2^5$, $f_{RES}/2^6$, $f_{RES}/2^7$)
- **Free-Run timer : 16 bits x 12 channels**
 - 16-bit free running counter, signals an interrupt when overflow or match with compare register
 - Includes prescaler ($f_{RES}/2^2$, $f_{RES}/2^4$, $f_{RES}/2^5$, $f_{RES}/2^6$)
 - Timer data register has R/W access
- **PPG : 16 bit x 32 channels**
 - 16 bit down counter, cycle and duty setting registers
 - Interrupt at triggering, cycle or duty match
 - PWM operation and one-shot operation
 - Internal prescaler allows $f_{RES}/2^0$, $f_{RES}/2^2$, $f_{RES}/2^4$, $f_{RES}/2^6$ as counter clock
 - Can be triggered by software, reload timer or external trigger
 - Reload timer 0/1 available as trigger for PPG 0/1/2/3
 - Reload timer 2/3 available as trigger for PPG 4/5/6/7
 - Reload timer 4/5 available as trigger for PPG 8/9/10/11
 - Reload timer 6/7 available as trigger for PPG 12/13/14/15
 - Reload timer 8 available as trigger for PPG 16/17
 - Reload timer 9 available as trigger for PPG 18/19
 - Reload timer 10 available as trigger for PPG 20/21
 - Reload timer 11 available as trigger for PPG 22/23
 - Reload timer 12 available as trigger for PPG 24/25
 - Reload timer 13 available as trigger for PPG 26/27
 - Reload timer 14 available as trigger for PPG 28/29
 - Reload timer 15 available as trigger for PPG 30/31
 - External trigger for PPG 0/8/16/24 (shared)
 - External trigger for PPG 1/9/17/25 (shared)
 - External trigger for PPG 2/10/18/26 (shared)
 - External trigger for PPG 3/11/19/27 (shared)
 - External trigger for PPG 4/12/20/28 (shared)
 - External trigger for PPG 5/13/21/29 (shared)
 - External trigger for PPG 6/14/22/30 (shared)
 - External trigger for PPG 7/15/23/31 (shared)
- **Input capture : 16 bits x 10 channels**
 - Rising edge, falling edge or rising & falling edge sensitive
 - Free-Run timer 0 available as trigger for input capture 0/1
 - Free-Run timer 1 available as trigger for input capture 2/3
 - Free-Run timer 4 available as trigger for input capture 4/5
 - Free-Run timer 5 available as trigger for input capture 6/7

MB91460 Series

- Free-Run timer 8 available as trigger for input capture 8/9
- Free-Run timer 8-11 can be used for Autosar applications

- **Output compare : 16 bits x 8 channels**
 - Signals an interrupt when a match with one of 16-bit IO timer occurs
 - An output signal can be generated
 - Free-Run timer 2 available as trigger for output compare 0/1
 - Free-Run timer 3 available as trigger for output compare 2/3
 - Free-Run timer 6 available as trigger for output compare 4/5
 - Free-Run timer 7 available as trigger for output compare 6/7

- **LIN-USART (LIN = Local Interconnect Network): 16 channels**
 - Full-duplex double buffer system (4 ch with 16 byte RX/TX FIFO buffer each; on MB91FV460B all channels with FIFO)
 - With parity/without parity selectable
 - 1 or 2 stop bits selectable
 - 7 or 8 bits data length selectable
 - NRZ type transfer format
 - Asynchronous /synchronous communications selectable
 - Master-slave communication function (multiprocessor mode)
 - Dedicated baud rate prescaler is embedded in each channel
 - External clock is able to use as transfer clock
 - Parity error, frame error, and overrun error detecting functions
 - SPI compatible
 - LIN master and slave
 - LIN USART 0/8 and ICU 0 co-operate (for LIN sync field in slave mode)
 - LIN USART 1/9 and ICU 1 co-operate (for LIN sync field in slave mode)
 - LIN USART 2/10 and ICU 2 co-operate (for LIN sync field in slave mode)
 - LIN USART 3/11 and ICU 3 co-operate (for LIN sync field in slave mode)
 - LIN USART 4/12 and ICU 4 co-operate (for LIN sync field in slave mode)
 - LIN USART 5/13 and ICU 5 co-operate (for LIN sync field in slave mode)
 - LIN USART 6/14 and ICU 6 co-operate (for LIN sync field in slave mode)
 - LIN USART 7/15 and ICU 7 co-operate (for LIN sync field in slave mode)

- **CAN: 6 channels**
 - Supports CAN protocol version 2.0 part A and B
 - Bit rates up to 1 Mbit/s
 - Up to 128 message objects
 - Each message object has its own identifier mask
 - Programmable FIFO mode (concatenation of message objects)
 - Maskable interrupt
 - Disabled Automatic Retransmission mode for Time Triggered CAN applications
 - Programmable loop-back mode for self-test operation

- I²C (400kbit/s fast mode): 8 channels
 - Master or slave transmission
 - Arbitration function
 - Clock synchronization function
 - Slave address and general call address detect function
 - Transfer direction detect function
 - Start condition repeat generation and detection function
 - Bus error detect function
 - Compatible to I2C standard and fast mode specification (operation up to 400kHz, 10 bit addressing)
 - Includes clock divider functionality
 - SCL and SDA lines include optional noise filter. The noise filter allows the suppression of spikes in the range of 1 to 1.5 cycles of the resource clock (CLKP)
- APIX[®] ¹ controller:
 - Main link (120Mbit / 6Mbit): 1 channel
 - Sideband channel links (5Mbit / 6Mbit): 2 links
- FlexRay controller: up to 2 channels
 - Conformance with FlexRay protocol specification v2.1
 - Data rates of up to 10 Mbit/s on each channel
- Inter-IC sound bus (I²S): 10 channels
 - master or slave operation support
 - operation up 2.5 MBit/s
 - 3 - 16bit data length support
- MOST support (MediaLB for controlling an external MOST IC is integrated)
 - Digital interface to external MOST controller
 - Frame sync pattern support
 - Scalable data rate for streaming, packet, control, isochronous
 - System-broadcast channel for administration
 - Broadcast support for synchronous data
 - Supports 512 Fs, 15 channels
 - Contains local channel buffers: 32 bit * 2 K
 - Contains a 32 bit * 2 K FIFO buffer for between MediaLB and I²S.

1. APIX[®] is a registered mark of INOVA Semiconductors GmbH.

- **USB: USB 1.1, correspond to USB Full Speed**
 - Configurable endpoints
 - Supports control, bulk, interrupt and isochronous transfer
 - built-in FIFO for all endpoints
 - clock and data recovery
- **PFM (pulse frequency modulator) : 16 bits x 1 channel**
 - 16-bit reload timers for generating high/low pulse waveforms
 - Includes clock prescaler ($f_{RES}/2^1$, $f_{RES}/2^3$, $f_{RES}/2^5$, $f_{RES}/2^6$, $f_{RES}/2^7$)
- **Sound Generator : 1 channel**
 - 8-bit PWM signal is mixed with tone frequency from 16-bit reload counter
 - PWM clock by internal prescaler: $f_{RES}/2^0$, $f_{RES}/2^1$, $f_{RES}/2^2$, $f_{RES}/2^3$, $f_{RES}/2^4$
 - Tone frequency: PWM frequency / 2 / (reload value + 1)
- **Stepper Motor Controller : 6 channels**
 - Four high current outputs for each channel
 - Two synchronized 8/10-bit PWMs per channel
 - Internal prescaling for PMW clock: $f_{RES}/1$, $f_{RES}/4$, $f_{RES}/5$, $f_{RES}/6$, $f_{RES}/8$, $f_{RES}/10$, $f_{RES}/12$, $f_{RES}/16$
- **LCD controller**
 - 4 common / 40 segment
 - Display: Up to 160 cells (for 1/4 duty cycle)
 - Duty cycle: Selectable from options: 1/2, 1/3 and 1/4
 - Bias: Fixed at 1/3
 - Frame period: Selectable from four options. (for clock, peripheral clock or Sub clock is selectable)
 - Driver: Built-in (for internal divided resistors), or external divided resistors can be connected to the V0-V3 pins
 - Data memory: Built-in 16-byte data memory for display
 - STOP state: Enable LCD display in the Sub-STOP state
 - Blank display: Selectable
 - Pin: The SEG0-39 of COM0-4 pin usage can be switched between general and specialized purposes
 - Others: External divided resistors can be also used to shut off the current when LCD is deactivated
- **Timebase/watchdog timer (26 bits)**
 - Adjustable watchdog timer interval (between 2^{20} and 2^{26} system clock cycles)
- **RC oscillator watchdog timer ("Hardware Watchdog", 16 bits)**
- **Real-time clock**
 - RTC module can be clocked either from 32kHz crystal, 4MHz quartz or from the RC Oscillator
 - Facility to correct oscillation deviation (Sub clock calibration)
 - Read/write accessible second/minute/ hour registers

- Can signal interrupts every half second/second/ minute/hour/day
- Internal clock divider and prescaler provide exact 1s clock based on a 4MHz or a 32kHz clock input
- Prescaler value for 4MHz is 0F423F_H
- Prescaler value for 32kHz is 001FFF_H
- Programmable CRC calculation module
 - Polynom and initial seed programmable
 - data to be calculated can be written with DMA
 - interrupt when calculation is finished
- Clock monitor (clock output function): 1 channel
- Clock supervisor
 - Monitors external 32kHz and 4MHz for fails (e.g. crystal breaks)
 - Switches in case of fail to an available recovery clock (CLKSUB or CLKRC)
- Supply supervisor (low voltage detection)
 - monitors external power supply and internal regulator output for low voltage
 - generate interrupt or reset if low voltage is detected
- Clock modulator (reduction of EME)
- Sub clock calibration
 - Calibration of the RTC timer in 32kHz or RC oscillator operation, based on the more accurate 4MHz quartz is possible
- Main oscillation stabilization timer
 - 23-bit counter for main oscillation stabilization wait when running in Sub clock mode
 - Generates an interrupt when stabilization time has elapsed
- Sub oscillation stabilization timer
 - 15-bit counter for sub oscillation stabilization wait when running in Main clock mode
 - Generates an interrupt when stabilization time has elapsed
- Low power consumption modes: SLEEP/Sub-RUN/RTC/STOP/ShutDown function

MB91460 Series

2.3. MB91460 Series Product Lineup

2.3.1 Evaluation Device Lineup MB91V460A, MB91FV460B

The product lineup table compares MB91V460A with the newer MB91FV460B.

Table 2.3-1 Evaluation device lineup MB91V460A and MB91FV460B

Feature	MB91V460A	MB91FV460B
Max. core frequency (CLKB)	80MHz	100MHz
Max. resource frequency (CLKP)	40MHz	50MHz
Max. external bus freq. (CLKT)	40MHz	50MHz
Max. CAN frequency (CLKCAN)	20MHz	20MHz
Max. FlexRay frequency (SCLK)	-	80MHz
Technology	0.35um	0.18um
Flash memory	Emulation SRAM 32bit read data	Internal Flash memory 2112KB + external emulation SRAM with 64bit read data
Satellite Flash	-	Data Flash 64 KByte
Flash Protection	-	yes, for main flash and data flash
CRC calculation	-	yes, fixed polynom
Programmable CRC calculation	-	yes, programmable polynom
D-bus RAM	64 KByte	64 KByte
I/D-bus RAM	64 KByte	64 KByte
Flash-cache (F-cache)	16 KByte	16 KByte
External bus cache (I-cache)	4 KBytes	4 KBytes
Standby RAM ¹	-	-
Boot-ROM	4 KByte fixed	16 KByte Boot Flash + 1KB Boot ROM ²
MMU/MPU ³	MPU (16 ch)	MPU (16 ch)
DMA	5 ch, 128 sources	5 ch, 256 sources
Software-Watchdog	yes	yes, with RUN flag
Hardware-Watchdog (RC osc. based)	yes (disengageable)	yes (disengageable), can be activated in SLEEP/STOP
Bit Search	yes	yes
RTC	1 ch	1 ch
Free Running Timer	8 ch	12 ch ⁴
ICU	8 ch	10 ch
OCU	8 ch	8 ch
Reload Timer	8 ch	16 ch, can be cascaded to 32bit
PPG 16-bit	16 ch	32 ch, the settings register are readable
PFM 16-bit	1 ch	1 ch
Sound Generator	1 ch	1 ch

Feature	MB91V460A	MB91FV460B
Up/Down Counter (8/16-bit)	4 ch (8-bit) / 2 ch (16-bit)	4 ch (8-bit) / 2 ch (16-bit)
SMC	6 ch	6 ch
LCD controller (40x4)	1 ch	1 ch, frame rate changed for higher clock speed
C_CAN	6 ch (128msg)	6 ch (128msg)
LIN-USART	4 ch + 4 ch FIFO + 8 ch	16 ch FIFO + End of Transmission IRQ
I2C (400k)	4 ch	8 ch
I ² S	-	10 ch
MediaLB (MOST interface)	-	1 ch (256Fs / 512Fs)
USB	-	USB 1.1 Full Speed, function + mini host
FlexRay	-	2 ch (A+B)
APIX [®] 5	-	1 PHY, 2 buffer
FR external bus	yes (32bit addr, 32bit data)	yes (32bit addr, 32bit data), separate supply
External Interrupts	16 ch	32 ch + interrupt input relocation
NMI Interrupts	1 ch	1 ch
General IO ports	288 (23 non-multiplexed)	328 (24 non-multiplexed)
IO pin relocation	-	MB91460B, P, T, M and Q series IO relocation
ADC (10 bit)	32 ch	32 ch ADC0 + 22 ch ADC1 ⁶
DAC (10 bit)	2 ch	2 ch
Alarm Comparator	2 ch	2 ch
ADC range comparator (digital)	-	8 ch (4 ch per ADC macro)
Reset input (INITX)	yes	yes
Hardware Standby input (HSTX)	yes	yes
Clock Modulator	yes	yes
Low Power Mode	yes	yes
Supply Supervisor	yes	yes
Clock Supervisor	yes	yes
Main clock oscillator	4MHz	4MHz
Sub clock oscillator	32kHz	32kHz
RC Oscillator	100kHz	100kHz / 2MHz
PLL	x 20	x 25
DSU4	yes	yes
EDSU	yes (32 BP)	yes (32 BP)
Debug kernel in BootROM	-	yes
Supply Voltage	3V / 5V	1.8V / 3V / 5V
Regulator	internal	control outputs for external regulator
Temperatur Range (Ta)	0..70 C	0..70 C

MB91460 Series

Feature	MB91V460A	MB91FV460B
Package	BG-660P-M02	BGA-896P-M02
Power on to PLL run	< 20 ms	< 20 ms
Flash Download Time	n.a.	< 8 sec

1. The Standby RAM of MB91F467E can be emulated with external S-RAM
2. ROM only for serial programming mode, used if the Boot Flash is empty.
3. The Memory protection Unit (MPU) is a part of the EDSU functionality
4. FRT8 can be used for ICU8+9, FRT8-11 can be used for Autosar
5. APIX[®] is a registered mark of INOVA Semiconductors GmbH.
6. 2 ADC macros with channel result registers;
ADC operation does not switch the attached GP port to input direction

2.3.2 MB91460 Series Device Lineup

The following table gives a short lineup of the MB91460 series devices.

Table 2.3-2 MB91460 series device lineup

Device	Package	Flash KByte	RAM KByte	Cache KByte	Freq. MHz	CAN ¹ ch.	Ext. Bus	Extras	Application
MB91V460A	BGA-660	-	128	16+4 ²	80	6 ³	32A 32D	Old eval. device	Evaluation
MB91FV460B	BGA-896	2112			100			New eval. device	Evaluation
MB91F464Ax	LQFP-100	416	16	-	80	1	-		Body control
MB91F464BB	LQFP-144	416	40	8	100	3	22A 16D	Port relocation: External Bus or resource I/O	Body control
MB91F465BB	LQFP-144	544							Body control
MB91F466Bx	LQFP-144	832			96	6			Body control
MB91F467Bx	LQFP-144	1088							Body control
MB91F463CA	LQFP-144	288	24	4	100	3	-	SMC 6 ch	Dashboard
MB91F465CA	LQFP-144	544	32	8					Dashboard
MB91F467Cx	LQFP-144	1088	64	8					Dashboard
MB91F465DA	QFP-208	544	48	8	100	3	26A 32D	SMC 6 ch	Dashboard
MB91F467Dx	QFP-208	1088	64		96				Dashboard
MB91F467EA	QFP-208	1088	128	8	100	2	26A 32D	SMC 6 ch, low leakage	Dashboard
MB91F469Gx	BGA-320	2112	96	16+4 ⁴	100	6	28A 32D	CAN 128 msg. buffers	Gateway
MB91F464Hx	LQFP-144	416	32	8	100	1	22A 16D		Body control
MB91F466HA	LQFP-144	832	40	8	96				Body control
MB91F465Kx	LQFP-120	544	16	4	80	1	-		Body control
MB91F467MA	LQFP-216	1088	64	8+4 ⁵	90	2 ⁶	24A 16D	Media LB + I ² S	Infotainment
MB91F463Nx	LQFP-64	288	10	4	100	2	-	low pin count	Body control

Device	Package	Flash KByte	RAM KByte	Cache KByte	Freq. MHz	CAN ¹ ch.	Ext. Bus	Extras	Application
MB91F465PA	LQFP-176	544	40	8	100	3	24A 16D	PPG 32 channels, port relocation	Body control
MB91F467PA	LQFP-176	1088	80			4			Body control
MB91F469QA	BGA-320	2112	96	16+4 ⁷	100	3	28A 32D	2 ADCs (32 + 8 ch)	Gateway
MB91F467Rx	LQFP-176	1088	64	8	90	2	24A 16D	CAN 32+64 msg.buffers	Infotainment
MB91F467Sx	LQFP-176	1088	64	8	100	2	24A 16D	APIX [®] 8	APIX [®]
MB91F467TA	LQFP-144	1088	64	8	100	2	24A 16D		Infotainment
MB91F469TA	LQFP-144	2112	128	16					Infotainment
MB91F465XA	LQFP-100	544	32	8	100	8	-	Flex Ray	Flex Ray

1. CAN with 32 message buffers, if not mentioned otherwise
2. 16 KByte Flash-Cache + 4 KByte Instruction-Cache on external bus
3. CAN with 128 message buffers
4. 16 KByte Flash-Cache + 4 KByte Instruction-Cache on external bus
5. 8 KByte Flash-Cache + 4 KByte Instruction-Cache on external bus
6. CAN with 32 + 64 message buffers
7. 16 KByte Flash-Cache + 4 KByte Instruction-Cache on external bus
8. APIX[®] is a registered mark of INOVA Semiconductors GmbH.

MB91460 Series

2.3.3 MB91460 Series Resource Lineup

The following table gives a short lineup of the standard resources implemented in MB91460 series devices. Note that the resource macro numbers are listed. For example, MB91460A series has CAN4 implemented, MB91460H-series has CAN0 and MB91460C-series has CAN0, CAN1, CAN2.

Table 2.3-3 MB91460 series resource lineup

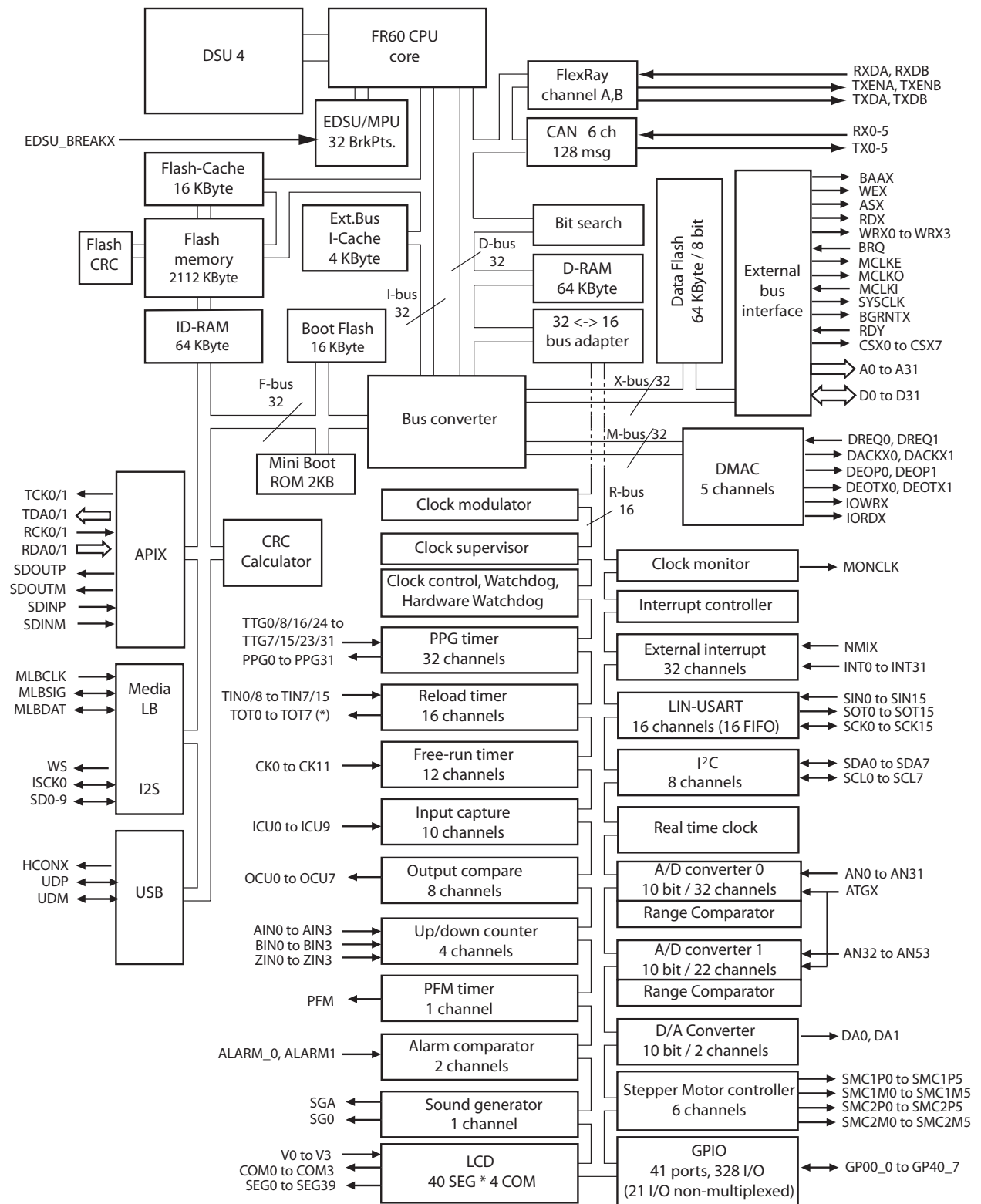
Resources	V460A	FV460B	MB91460 Series															
			A	B	C	D	E	G	H	K	M	N	P	Q	R	S	T	X
CAN	0-5	0-5	4	0-2 3-5 ¹	0-2	0-2	0,1	0-5	0	4	0,1	4,5	0-2, 3 ²	0,1,5	0,1	0,1	0,1	0,4
PPG	0-15	0-31	0-9	0-15	4-15	4-15	4-15	0-15	0-15	0-11	0-7	0-7	0-31	0-15	0-7	0-15	0-5 8-15	0-11
LIN- USART	0-15	0-15	0-4	0, 2-7	2, 4-7	2, 4-7	2, 4-7	0-7	0, 2-7	0-4	0-8	0-3	0-11	0-11	0-6	2-7	0, 2-11	4,6,7
I ² C	0-3	0-7	0	0,1	0,2,3	0,2,3	0,2,3	0-3	0,1	0	0-7	2,3	0-3	1-3	0-2	0-2	0-3	0
FRT	0-7	0-11	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-3	0-3	0-7	0-8	0-3	0-7	0-7	0-7
ICU	0-7	0-9	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-3	0-3	0-7	0-9	0-3	0-7	0-7	0-7
OCU	0-7	0-7	0-5	0-7	0-3	0-3	0-3	0-7	0-7	0-7	0-3	0-3	0-7	0-7	0-3	0-3	0-7	0-5
RLT	0-7	0-15	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0-3,7	0-3,7	0-15	0-7	0-3,7	0-7	0-7	0-7
Ext. INT	0-15, NMI	0-31, NMI	0-7, 12,14	0-15, NMI	0-14, NMI	0-10, 12,14	0-10, 12,14	0-15	0-15, NMI	0-7, 12,14	0-15	0-7, 12,13	0-15, NMI	0-31	0-15, NMI	0-15, NMI	0-9, 14,15, NMI	0-8, 14,15
UDC	0-3	0-3	-	0,1	0,2,3	0,2,3	0,2,3	0-3	0,1	-	-	0,1	0-3	0-3	-	0-3	0-3	-
PFM	0	0	-	-	0	0	0	0	-	-	0	-	0	0	0	0	0	-
Sound	0	0	-	0	0	0	0	0	0	0	-	-	0	0	-	0	0	-
RTC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SMC	0-5	0-5	-	-	0-5	0-5	0-5	-	-	-	-	-	-	-	-	-	-	-
ADC	0	0,1 RCO ³	0	0	0	0	0 RCO ⁴	0	0	0	0	0	0,1 ⁵ RCO ⁶	0,1	0	0	0	0
ADC chan.	0-31	0-31 32- 53 ⁷	0-12 16-23	0-31	0-13 16-31	0-7 16-31	0-7 16-31	0-31	0-31	0-25	0-11	0-7	0-31 37- 46 ⁸	0-31, 34-39 42, 46 ⁹	0-15	0-15	0-31	0-12 16-19
ALARM	0,1	0,1	-	0	0	0	0	0,1	0	-	-	-	-	0,1	-	0	-	-
DAC	0,1	0,1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LCD	4x40	4x40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1. CAN3-5 available on MB91F466Bx and MB91F467Bx only
2. CAN3 available on MB91F467P only
3. RCo: ADC is equipped with the Range Comparator
4. RCo: ADC is equipped with the Range Comparator
5. ADC1 available on MB91F467P only
6. RCo: ADC is equipped with the Range Comparator on MB91F467Px only
7. ADC channels 32-53 connected to macro ADC1
8. ADC channels 37-46 available on MB91F467Px only, connected to macro ADC1
9. ADC channels 34-39, 42, 46 connected to macro ADC1

2.4. Block Diagram

The following illustration shows the block diagram of MB91FV460B.

Figure 2.4-1 Block Diagram MB91FV460B



Notes: (*) Reload timer 8-15 without external output pin

MB91460 Series

2.5. Debug Support Features

This section describes the debug support features of MB91FV460B in comparison with MB91V460A.

- MB91460 Series Modes
- Embedded Program/Data Memory (Flash)
- External Emulation S-RAM interface with 64 bit data width
- Debug Support Unit (DSU4)
- Extended Real Time Monitor
- Embedded Debug Support / Memory Protection Unit (EDSU/MPU)
- ROM Select Register (ROMS)
- External Operation Control Pins

2.5.1 MB91460 Series Modes

MB91FV460B can emulate all currently available or specified MB91460 series devices. Because some devices have non-standard features (e.g. multiplexing of pin functions, re-location of interrupts and analog channels, and others), MB91FV460B can work in different “Series Modes”, controlled by 3 external pins. For details, see section [3.8.1 MB91460 Series Modes \(Page No.143\)](#).

2.5.2 Embedded Program/Data Memory (Flash)

MB91FV460B has a 2112 KByte Flash memory which can be used the same kind as in normal MB91460 series flash devices. For the flash description, please refer to chapter [Chapter 54 Flash Memory \(Page No.1175\)](#).

2.5.3 External Emulation S-RAM interface with 64-bit data width

The interface for the external emulation S-RAM has been enhanced for 64-bit data width. So it is possible to emulate 64-bit flash access with prefetch. If the external S-RAM is used, all 64 data bit lines must beconnected to S-RAM on the evaluation board. The read/write operations from/to external emulation S-RAM (via DSU4) are not changed.

For the pin description of the S-RAM interface, please refer to section [3.8.8 Dedicated and Special Pins \(Page No.165\)](#). Information about the selection between external emulation S-RAM and internal Flash memory is given in following section [2.5.4 ROM Select Register \(ROMS\) \(Page No.31\)](#).

2.5.4 ROM Select Register (ROMS)

Like on MB91V460A, the ROMS register can be read and written by the CPU.

The following table lists the differences of MB91FV460B's ROMS register versus MB91V460A:

Item	MB91V460A	MB91FV460B
Initialization by...	Settings initialization (INIT)	Power On Reset
Initialization value	0xFF80; ROMS[6:0]=0 enables an (external) 1MByte flash	0xF800; ROMS[10:0]=0 enables the internal flash 2MB

The access to the internal flash is controlled by the setting of the ROMS[10:0] bits and the pin SRAM_SFX. The ROMS[15:0] register switches the access between Flash/S-RAM and external bus.

● ROMS Register

ROMS: ROM Select Register Address: 0x0390 Access: read/write half-word (16-bit)

15	14	13	12	11	10	9	8	Bit
ROMS15	ROMS14	ROMS13	ROMS12	ROMS11	ROMS10	ROMS09	ROMS08	
1	1	1	1	1	0	0	0	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute

7	6	5	4	3	2	1	0	Bit
ROMS07	ROMS06	ROMS05	ROMS04	ROMS03	ROMS02	ROMS01	ROMS00	
0	0	0	0	0	0	0	0	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute

ROMSn	Function
0	Access to the address range controlled by ROMSn goes to the internal flash memory or the external emulation SRAM, depending on SRAM_SFX setting.
1	Access to the address range controlled by ROMSn goes to the external bus.

For the ROMSn address range assignment, please see the IO MAP starting at address [040000H \(Page No.109\)](#). Per default on MB91FV460B, ROMS11 to ROMS15 point to external Bus and ROMS00 to ROMS10 point to Flash/SRAM, covering the size of the 2 MByte internal flash.

● SRAM_SFX pin

If ROMSn points to Flash/S-RAM, the control pin SRAM_SFX switches between internal flash and external emulation S-RAM.

SRAM_SFX	Function
0	The internal flash is enabled and can be accessed like on other MB91460 series devices. The external Emulation SRAM is disabled.
1	The internal flash is disabled. All access to the program/data memory area is switched to the external Emulation SRAM.

The external Emulation S-RAM covers the address range from 0x040000 to 0x24FFFF. Please clear the attached bits of ROMS register to enable S-RAM access. See the IO MAP starting at address [040000H \(Page No.109\)](#) for ROMS address range assignment.

It is not possible to mix internal Flash acces with external Emulation S-RAM acces.

MB91460 Series

2.5.5 Debug Support Unit (DSU4)

The Debug Support Unit DSU4 is the same as used on MB91V460A.

2.5.6 Extended Real Time Monitor (RTM)

On MB91V460A, only the Real Time Monitor output pins IHIT_3 to IHIT_0 (Instruction address match) were available.

On MB91FV460B, the following Real Time Monitor outputs have been added:

- DHIT_5 to DHIT_0: Operand address / data match
- MDD_31 to MDD_0: Operand data bus (data from internal D-Bus)
- MDDEN_3 to MDDEN_0: Operand data bus write strobes
- MRSTX, RCLK: Real time monitor reset and clock

For the pin list, see chapter PIN DESCRIPTION, section “Dedicated and Special Pins” on page 34.

The RTM module has been slightly changed: The trace output on MDDEN[3:0], MDD[31:0] starts after the RTM control register has been written for the first time. The trace output is disabled by Tool Reset (TRSTX).

2.5.7 Embedded Debug Support / Memory Protection Unit

MB91FV460B has the standard 460 series EDSU/MPU with 8 groups / 32 channels.

Please refer to chapter [Chapter 29 MPU / EDSU \(Page No.531\)](#).

2.5.8 External Operation Control Pins

The following external input pins are used to control the operation:

● Fast Clock Input

FCI	Fast clock input
0	fast clock input mode disabled
1	fast clock input mode enabled, max. frequency on X0 is 16 MHz

● Flash Security Disable

FSC_DISABLE	Flash Security disable
0	Flash Security is ON
1	Flash Security is OFF, does not fetch security information from flash. All read and write access are always enabled *1

1. After INIT, please wait 3 ms before accessing the Data Flash macro.
This time is needed for internal data flash clock synchronisation.

● Clock Supervisor Kill

CSV_KILL	Clock Supervisor kill
0	Clock Supervisor is ON
1	Clock Supervisor is OFF

● Hardware Watchdog Kill

HWWDG_KILL	Hardware Watchdog kill
0	Hardware Watchdog is ON
1	Hardware Watchdog is OFF

● Fixed Reset Vector Enable

FIX_ENX	Fixed Mode/Reset Vector enable (low active)
0	Device bootes using fixed mode and reset vector
1	Device bootes using mode and reset vector from flash data

MB91460 Series

Chapter 3 MB91460 Series Basic Information

This chapter describes MB91460 series basic information including Memory- and I/O map, interrupt vector table, pin function list, circuit type and pin state table for each device mode.

3.1. Memory Map

Figure 3.1-1 Memory Map

MB91V460A		MB91FV460B	
00000000 _H	I/O (direct addressing area)	00000000 _H	I/O (direct addressing area)
00000400 _H	I/O	00000400 _H	I/O
00001000 _H	DMA	00001000 _H	DMA
00002000 _H	Instruction RAM / Flash-Cache (16 KBytes)	00002000 _H	Instruction RAM / Flash-Cache (16 KBytes)
00006000 _H		00006000 _H	Media LB / I2S
00007000 _H	Flash memory control	00007000 _H	Flash memory control
00007200 _H		00007200 _H	APIX
		00007400 _H	USB
		00007500 _H	
		00007800 _H	Mini Boot ROM (2 KBytes)
0000B000 _H	Boot ROM (4 KBytes)	00008000 _H	Boot Flash (16 KBytes)
0000C000 _H	CAN	0000C000 _H	CAN
0000D000 _H		0000D000 _H	FlexRay
		0000D800 _H	
0000F000 _H	EDSU / MPU	0000F000 _H	EDSU / MPU
0000F100 _H		0000F100 _H	
00010000 _H	2 Way Set Associative I-Cache 4 KB	00010000 _H	2 Way Set Associative I-Cache 4 KB
00020000 _H	D-RAM (0 wait, 64 KB)	00020000 _H	D-RAM (0 wait, 64 KB)
00030000 _H	I/D-RAM (64 KB)	00030000 _H	I/D-RAM (64 KB)
00040000 _H	External emulation S-RAM	00040000 _H	Internal emulation flash memory (2112 KB) / External emulation S-RAM (selectable internal or external)
00280000 _H	External emulation S-RAM / External bus area	00280000 _H	External emulation S-RAM / External bus area
00500000 _H	External data bus	00500000 _H	External data bus
		FFFBF000 _H FFBFF000 _H	
		FFFCFFFF _H FFFFFFF _H	Data Flash 64KB + 256Byte / External bus area

Note: Access prohibited areas

- Table convention

MSB LSB

Leftmost register address
(For Word access, first register becomes MSB side of the data.)

MB91460 Series**Table 3.2-1 I/O Map**

Address	Register				Block
	+0	+1	+2	+3	
000000 _H	PDR00 [R/W] XXXXXXXX	PDR01 [R/W] XXXXXXXX	PDR02 [R/W] XXXXXXXX	PDR03 [R/W] XXXXXXXX	General Purpose IO Port Data Register
000004 _H	PDR04 [R/W] XXXXXXXX	PDR05 [R/W] XXXXXXXX	PDR06 [R/W] XXXXXXXX	PDR07 [R/W] XXXXXXXX	
000008 _H	PDR08 [R/W] XXXXXXXX	PDR09 [R/W] XXXXXXXX	PDR10 [R/W] XXXXXXXX	PDR11 [R/W] XXXXXXXX	
00000C _H	PDR12 [R/W] XXXXXXXX	PDR13 [R/W] XXXXXXXX	PDR14 [R/W] XXXXXXXX	PDR15 [R/W] XXXXXXXX	
000010 _H	PDR16 [R/W] XXXXXXXX	PDR17 [R/W] XXXXXXXX	PDR18 [R/W] XXXXXXXX	PDR19 [R/W] XXXXXXXX	
000014 _H	PDR20 [R/W] XXXXXXXX	PDR21 [R/W] XXXXXXXX	PDR22 [R/W] XXXXXXXX	PDR23 [R/W] XXXXXXXX	
000018 _H	PDR24 [R/W] XXXXXXXX	PDR25 [R/W] XXXXXXXX	PDR26 [R/W] XXXXXXXX	PDR27 [R/W] XXXXXXXX	
00001C _H	PDR28 [R/W] XXXXXXXX	PDR29 [R/W] XXXXXXXX	PDR30 [R/W] XXXXXXXX	PDR31 [R/W] XXXXXXXX	
000020 _H	PDR32 [R/W] XXXXXXXX	PDR33 [R/W] XXXXXXXX	PDR34 [R/W] XXXXXXXX	PDR35 [R/W] XXXXXXXX	
000024 _H	PDR36 [R/W] XXXXXXXX	PDR37 [R/W] XXXXXXXX	PDR38 [R/W] XXXXXXXX	PDR39 [R/W] XXXXXXXX	
000028 _H	PDR40 [R/W] XXXXXXXX	reserved	reserved	reserved	
00002C _H	reserved				reserved
000030 _H	EIRR0 [R/W] XXXXXXXX	ENIR0 [R/W] 00000000	ELVR0 [R/W] 00000000 00000000		Ext. INT 0-7 NMI
000034 _H	EIRR1 [R/W] XXXXXXXX	ENIR1 [R/W] 00000000	ELVR1 [R/W] 00000000 00000000		Ext. INT 8-15
000038 _H	DICR [R/W] ----- 0	HRCL [R/W] 0 - - 11111	RBSYNC ¹		DLYI/I-unit
00003C _H	RTEST0 [R/W] XXXXXXXX	RTEST1 [R/W] XXXXXXXX	RTEST2 [R/W] XXXXXXXX	RTEST3 [R/W] XXXXXXXX	R-Bus Testregister (hidden)
000040 _H	SCR00 [R/W,W] 00000000	SMR00 [R/W,W] 00000000	SSR00 [R/W,R] 00001000	RDR00/TDR00 [R/W] 00000000	LIN-USART 0 with FIFO
000044 _H	ESCR00 [R/W] 00000X00	ECCR00 [R/W,R,W] 000000XX	FSR00 [R/W,R] XX00 0000	FCR00 [R/W] 0001 0000	

Address	Register				Block
	+0	+1	+2	+3	
000048 _H	SCR01 [R/W,W] 00000000	SMR01 [R/W,W] 00000000	SSR01 [R/W,R] 00001000	RDR01/TDR01 [R/W] 00000000	LIN-USART 1 with FIFO
00004C _H	ESCR01 [R/W] 00000X00	ECCR01 [R/W,R,W] 000000XX	FSR01 [R/W,R] XX00 0000	FCR01 [R/W] 0001 0000	
000050 _H	SCR02 [R/W,W] 00000000	SMR02 [R/W,W] 00000000	SSR02 [R/W,R] 00001000	RDR02/TDR02 [R/W] 00000000	LIN-USART 2 with FIFO
000054 _H	ESCR02 [R/W] 00000X00	ECCR02 [R/W,R,W] 000000XX	FSR02 [R/W,R] XX00 0000	FCR02 [R/W] 0001 0000	
000058 _H	SCR03[R/W,W] 00000000	SMR03 [R/W,W] 00000000	SSR03 [R/W,R] 00001000	RDR03/TDR02 [R/W] 00000000	LIN-USART 3 with FIFO
00005C _H	ESCR03 [R/W] 00000X00	ECCR03 [R/W,R,W] 000000XX	FSR03 [R/W,R] XX00 0000	FCR03 [R/W] 0001 0000	
000060 _H	SCR04 [R/W,W] 00000000	SMR04 [R/W,W] 00000000	SSR04 [R/W,R] 00001000	RDR04/TDR04 [R/W] 00000000	LIN-USART 4 with FIFO
000064 _H	ESCR04 [R/W] 00000X00	ECCR04 [R/W,R,W] 000000XX	FSR04 [R/W,R] XX00 0000	FCR04 [R/W] 0001 0000	
000068 _H	SCR05 [R/W,W] 00000000	SMR05 [R/W,W] 00000000	SSR05 [R/W,R] 00001000	RDR05/TDR05 [R/W] 00000000	LIN-USART 5 with FIFO
00006C _H	ESCR05 [R/W] 00000X00	ECCR05 [R/W,R,W] 000000XX	FSR05 [R/W,R] XX00 0000	FCR05 [R/W] 0001 0000	
000070 _H	SCR06 [R/W,W] 00000000	SMR06 [R/W,W] 00000000	SSR06 [R/W,R] 00001000	RDR06/TDR06 [R/W] 00000000	LIN-USART 6 with FIFO
000074 _H	ESCR06 [R/W] 00000X00	ECCR06 [R/W,R,W] 000000XX	FSR06 [R/W,R] XX00 0000	FCR06 [R/W] 0001 0000	
000078 _H	SCR07 [R/W,W] 00000000	SMR07 [R/W,W] 00000000	SSR07 [R/W,R] 00001000	RDR07/TDR07 [R/W] 00000000	LIN-USART 7 with FIFO
00007C _H	ESCR07 [R/W] 00000X00	ECCR07 [R/W,R,W] 000000XX	FSR07[R/W,R] XX00 0000	FCR07 [R/W] 0001 0000	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000080 _H	BGR100 [R/W] 00000000	BGR000 [R/W] 00000000	BGR101 [R/W] 00000000	BGR001 [R/W] 00000000	Baudrate Generator LIN-USART 0-7
000084 _H	BGR102 [R/W] 00000000	BGR002 [R/W] 00000000	BGR103 [R/W] 00000000	BGR003 [R/W] 00000000	
000088 _H	BGR104 [R/W] 00000000	BGR004 [R/W] 00000000	BGR105 [R/W] 00000000	BGR005 [R/W] 00000000	
00008C _H	BGR106 [R/W] 00000000	BGR006 [R/W] 00000000	BGR107 [R/W] 00000000	BGR007 [R/W] 00000000	
000090 _H	PWC20 [R/W] ----- XX XXXXXXXX		PWC10 [R/W] ----- XX XXXXXXXX		Stepper Motor 0
000094 _H	reserved	reserved	PWS20 [R/W] -0000000	PWS10 [R/W] - -000000	
000098 _H	PWC21 [R/W] ----- XX XXXXXXXX		PWC11 [R/W] ----- XX XXXXXXXX		Stepper Motor 1
00009C _H	reserved	reserved	PWS21 [R/W] -0000000	PWS11 [R/W] - -000000	
0000A0 _H	PWC22 [R/W] ----- XX XXXXXXXX		PWC12 [R/W] ----- XX XXXXXXXX		Stepper Motor 2
0000A4 _H	reserved	reserved	PWS22 [R/W] -0000000	PWS12 [R/W] - -000000	
0000A8 _H	PWC23 [R/W] ----- XX XXXXXXXX		PWC13 [R/W] ----- XX XXXXXXXX		Stepper Motor 3
0000AC _H	reserved	reserved	PWS23 [R/W] -0000000	PWS13 [R/W] - -000000	
0000B0 _H	PWC24 [R/W] ----- XX XXXXXXXX		PWC14 [R/W] ----- XX XXXXXXXX		Stepper Motor 4
0000B4 _H	reserved	reserved	PWS24 [R/W] -0000000	PWS14 [R/W] - -000000	
0000B8 _H	PWC25 [R/W] ----- XX XXXXXXXX		PWC15 [R/W] ----- XX XXXXXXXX		Stepper Motor5
0000BC _H	reserved	reserved	PWS25 [R/W] -0000000	PWS15 [R/W] - -000000	
0000C0 _H	reserved	PWC0 [R/W] -00000--	reserved	PWC1 [R/W] -00000--	Stepper Motor Control 0-5
0000C4 _H	reserved	PWC2 [R/W] -00000--	reserved	PWC3 [R/W] -00000--	
0000C8 _H	reserved	PWC4 [R/W] -00000--	reserved	PWC5 [R/W] -00000--	
0000CC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
0000D0 _H	IBCR0 [R/W] 00000000	IBSR0 [R] 00000000	ITBAH0 [R/W] ----- 00	ITBAL0 [R/W] 00000000	I2C 0
0000D4 _H	ITMKH0 [R/W] 00 ---- 11	ITMKL0 [R/W] 11111111	ISMK0 [R/W] 01111111	ISBA0 [R/W] - 0000000	
0000D8 _H	reserved	IDAR0 [R/W] 00000000	ICCR0 [R/W] 00011111	reserved	
0000DC _H	IBCR1 [R/W] 00000000	IBSR1 [R] 00000000	ITBAH1 [R/W] ----- 00	ITBAL1 [R/W] 00000000	I2C 1
0000E0 _H	ITMKH1 [R/W] 00 ---- 11	ITMKL1 [R/W] 11111111	ISMK1 [R/W] 01111111	ISBA1 [R/W] - 0000000	
0000E4 _H	reserved	IDAR1 [R/W] 00000000	ICCR1 [R/W] 00011111	reserved	
0000E8 _H	LCDCMR [R/W] ---- 0000	LCR0 [R/W] 00010000	LCR1H [R/W] ----- 00	LCR1L [R/W] 00000000	LCD Controller
0000EC _H	VRAM00 [R/W] XXXXXXXX	VRAM01 [R/W] XXXXXXXX	VRAM02 [R/W] XXXXXXXX	VRAM03 [R/W] XXXXXXXX	
0000F0 _H	VRAM04 [R/W] XXXXXXXX	VRAM05 [R/W] XXXXXXXX	VRAM06 [R/W] XXXXXXXX	VRAM07 [R/W] XXXXXXXX	
0000F4 _H	VRAM08 [R/W] XXXXXXXX	VRAM09 [R/W] XXXXXXXX	VRAM10 [R/W] XXXXXXXX	VRAM11 [R/W] XXXXXXXX	
0000F8 _H	VRAM12 [R/W] XXXXXXXX	VRAM13 [R/W] XXXXXXXX	VRAM14 [R/W] XXXXXXXX	VRAM15 [R/W] XXXXXXXX	
0000FC _H	VRAM16 [R/W] XXXXXXXX	VRAM17 [R/W] XXXXXXXX	VRAM18 [R/W] XXXXXXXX	VRAM19 [R/W] XXXXXXXX	
000100 _H	GCN10 [R/W] 00110010 00010000		reserved	GCN20 [R/W] ---- 0000	PPG Control 0-3
000104 _H	GCN11 [R/W] 00110010 00010000		reserved	GCN21 [R/W] ---- 0000	PPG Control 4-7
000108 _H	GCN12 [R/W] 00110010 00010000		reserved	GCN22 [R/W] ---- 0000	PPG Control 8-11
00010C _H	reserved				
000110 _H	PTMR00 [R] 11111111 11111111		PCSR00 [R/W] XXXXXXXX XXXXXXXX		PPG 0
000114 _H	PDUT00 [R/W] XXXXXXXX XXXXXXXX		PCNH00 [R/W] 0000000 -	PCNL00 [R/W] 000000 - 0	
000118 _H	PTMR01 [R] 11111111 11111111		PCSR01 [R/W] XXXXXXXX XXXXXXXX		PPG 1
00011C _H	PDUT01 [R/W] XXXXXXXX XXXXXXXX		PCNH01 [R/W] 0000000 -	PCNL01 [R/W] 000000 - 0	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000120 _H	PTMR02 [R] 11111111 11111111		PCSR02 [R/W] XXXXXXXX XXXXXXXX		PPG 2
000124 _H	PDUT02 [R/W] XXXXXXXX XXXXXXXX		PCNH02 [R/W] 0000000 -	PCNL02 [R/W] 000000 - 0	
000128 _H	PTMR03 [R] 11111111 11111111		PCSR03 [R/W] XXXXXXXX XXXXXXXX		PPG 3
00012C _H	PDUT03 [R/W] XXXXXXXX XXXXXXXX		PCNH03 [R/W] 0000000 -	PCNL03 [R/W] 000000 - 0	
000130 _H	PTMR04 [R] 11111111 11111111		PCSR04 [R/W] XXXXXXXX XXXXXXXX		PPG 4
000134 _H	PDUT04 [R/W] XXXXXXXX XXXXXXXX		PCNH04 [R/W] 0000000 -	PCNL04 [R/W] 000000 - 0	
000138 _H	PTMR05 [R] 11111111 11111111		PCSR05 [R/W] XXXXXXXX XXXXXXXX		PPG 5
00013C _H	PDUT05 [R/W] XXXXXXXX XXXXXXXX		PCNH05 [R/W] 0000000 -	PCNL05 [R/W] 000000 - 0	
000140 _H	PTMR06 [R] 11111111 11111111		PCSR06 [R/W] XXXXXXXX XXXXXXXX		PPG 6
000144 _H	PDUT06 [R/W] XXXXXXXX XXXXXXXX		PCNH06 [R/W] 0000000 -	PCNL06 [R/W] 000000 - 0	
000148 _H	PTMR07 [R] 11111111 11111111		PCSR07 [R/W] XXXXXXXX XXXXXXXX		PPG 7
00014C _H	PDUT07 [R/W] XXXXXXXX XXXXXXXX		PCNH07 [R/W] 0000000 -	PCNL07 [R/W] 000000 - 0	
000150 _H	PTMR08 [R] 11111111 11111111		PCSR08 [R/W] XXXXXXXX XXXXXXXX		PPG 8
000154 _H	PDUT08 [R/W] XXXXXXXX XXXXXXXX		PCNH08 [R/W] 0000000 -	PCNL08 [R/W] 000000 - 0	
000158 _H	PTMR09 [R] 11111111 11111111		PCSR09 [R/W] XXXXXXXX XXXXXXXX		PPG 9
00015C _H	PDUT09 [R/W] XXXXXXXX XXXXXXXX		PCNH09 [R/W] 0000000 -	PCNL09 [R/W] 000000 - 0	
000160 _H	PTMR10 [R] 11111111 11111111		PCSR10 [R/W] XXXXXXXX XXXXXXXX		PPG 10
000164 _H	PDUT10 [R/W] XXXXXXXX XXXXXXXX		PCNH10 [R/W] 0000000 -	PCNL10 [R/W] 000000 - 0	
000168 _H	PTMR11 [R] 11111111 11111111		PCSR11 [R/W] XXXXXXXX XXXXXXXX		PPG 11
00016C _H	PDUT11 [R/W] XXXXXXXX XXXXXXXX		PCNH11 [R/W] 0000000 -	PCNL11 [R/W] 000000 - 0	

Address	Register				Block
	+0	+1	+2	+3	
000170 _H	P0TMCSRH [R/W] - 0 - 000 - 0	P0TMCSRL [R/W] 01000000	P1TMCSRH [R/W] - 0000000	P1TMCSRL [R/W] 01000000	Pulse Frequency Modu- lator
000174 _H	P0TMRLR [W] XXXXXXXXXX XXXXXXXXX		P0TMR [R] XXXXXXXXXX XXXXXXXXX		
000178 _H	P1TMRLR [W] XXXXXXXXXX XXXXXXXXX		P1TMR [R] XXXXXXXXXX XXXXXXXXX		
00017C _H	reserved				
000180 _H	reserved	ICS01 [R/W] 00000000	reserved	ICS23 [R/W] 00000000	Input Capture 0-3
000184 _H	IPCP0 [R] XXXXXXXXXX XXXXXXXXX		IPCP1 [R] XXXXXXXXXX XXXXXXXXX		
000188 _H	IPCP2 [R] XXXXXXXXXX XXXXXXXXX		IPCP3 [R] XXXXXXXXXX XXXXXXXXX		
00018C _H	OCS01 [R/W] - - - 0 - - 00 0000 - - 00		OCS23 [R/W] - - - 0 - - 00 0000 - - 00		Output Compare 0-3
000190 _H	OCCP0 [R/W] XXXXXXXXXX XXXXXXXXX		OCCP1 [R/W] XXXXXXXXXX XXXXXXXXX		
000194 _H	OCCP2 [R/W] XXXXXXXXXX XXXXXXXXX		OCCP3 [R/W] XXXXXXXXXX XXXXXXXXX		
000198 _H	SGCRH [R/W] 0000 - - 00	SGCRL [R/W] - - 0 - - 000	SGFR [R/W, R] XXXXXXXXXX XXXXXXXXX		Sound Generator
00019C _H	SGAR [R/W] 00000000	SGDAD [R/W] - - - - - 00	SGTR [R/W] XXXXXXXXXX	SGDR [R/W] XXXXXXXXXX	
0001A0 _H	AD0ERH [R/W] (ADERH [R/W]) 00000000 00000000		AD0ERL [R/W] (ADERL [R/W]) 00000000 00000000		A/D Converter 0 (in braces the old register names)
0001A4 _H	AD0CS1 [R/W] (ADCS1 [R/W]) 00000000	AD0CS0 [R/W] (ADCS0 [R/W]) 00000000	AD0CR1 [R] (ADCR1 [R]) 000000XX	AD0CR0 [R] (ADCR0 [R]) XXXXXXXXXX	
0001A8 _H	AD0CT1 [R/W] (ADCT1 [R/W]) 00010000	AD0CT0 [R/W] (ADCT0 [R/W]) 00101100	AD0SCH [R/W] (ADSCH [R/W]) - - - 00000	AD0ECH [R/W] (ADECH [R/W]) - - - 00000	
0001AC _H	reserved	ACSR0 [R/W] 011XXX00	reserved	ACSR1 [R/W] 011XXX00	Alarm Comparator 0-1
0001B0 _H	TMRLRC0 [W] XXXXXXXXXX XXXXXXXXX		TMRC0 [R] XXXXXXXXXX XXXXXXXXX		Reload Timer 0 (PPG 0-1)
0001B4 _H	reserved		TMCSRCH0 [R/W] - - - 00000	TMCSRCL0 [R/W] 0 - 000000	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0001B8 _H	TMRLRC1 [W] XXXXXXXXX XXXXXXXXX		TMRC1 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 1 (PPG 2-3)
0001BC _H	reserved		TMCSRCH1 [R/W] - - - 00000	TMCSRCL1 [R/W] 0 - 000000	
0001C0 _H	TMRLRC2 [W] XXXXXXXXX XXXXXXXXX		TMRC2 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 2 (PPG 4-5)
0001C4 _H	reserved		TMCSRCH2 [R/W] - - - 00000	TMCSRCL2 [R/W] 0 - 000000	
0001C8 _H	TMRLRC3 [W] XXXXXXXXX XXXXXXXXX		TMRC3 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 3 (PPG 6-7)
0001CC _H	reserved		TMCSRCH3 [R/W] - - - 00000	TMCSRCL3 [R/W] 0 - 000000	
0001D0 _H	TMRLRC4 [W] XXXXXXXXX XXXXXXXXX		TMRC4 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 4 (PPG 8-9)
0001D4 _H	reserved		TMCSRCH4 [R/W] - - - 00000	TMCSRCL4 [R/W] 0 - 000000	
0001D8 _H	TMRLRC5 [W] XXXXXXXXX XXXXXXXXX		TMRC5 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 5 (PPG10-11)
0001DC _H	reserved		TMCSRCH5 [R/W] - - - 00000	TMCSRCL5 [R/W] 0 - 000000	
0001E0 _H	TMRLRC6 [W] XXXXXXXXX XXXXXXXXX		TMRC6 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 6 (PPG 12-13)
0001E4 _H	reserved		TMCSRCH6 [R/W] - - - 00000	TMCSRCL6 [R/W] 0 - 000000	
0001E8 _H	TMRLRC7 [W] XXXXXXXXX XXXXXXXXX		TMRC7 [R] XXXXXXXXX XXXXXXXXX		Reload Timer 7 (PPG 14-15, A/D Converter)
0001EC _H	reserved		TMCSRCH7 [R/W] - - - 00000	TMCSRCL7 [R/W] 0 - 000000	
0001F0 _H	TCDT0 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS0 [R/W] 00000000	Free Running Timer 0 (ICU 0-1)
0001F4 _H	TCDT1 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS1 [R/W] 00000000	Free Running Timer 1 (ICU 2-3)
0001F8 _H	TCDT2 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS2 [R/W] 00000000	Free Running Timer 2 (OCU 0-1)

Address	Register				Block
	+0	+1	+2	+3	
0001FC _H	TCDT3 [R/W] XXXXXXXX XXXXXXXX		reserved	TCCS3 [R/W] 00000000	Free Running Timer 3 (OCU 2-3)
000200 _H	DMACA0 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC 0
000204 _H	DMACB0 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000208 _H	DMACA1 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC 1
00020C _H	DMACB1 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000210 _H	DMACA2 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC 2
000214 _H	DMACB2 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000218 _H	DMACA3 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC 3
00021C _H	DMACB3 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000220 _H	DMACA4 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC 4
000224 _H	DMACB4 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000228 _H - 00023C _H	reserved				
000240 _H	DMACR [R/W] 0 - - 0 0000	reserved			DMAC Control
000244 _H - 00024C _H	reserved				
000250 _H	DMATEST0 [R/W] XXXXXXXX 00000000 00000000 0000XXXX				DMAC Test Reg- ister (hidden)
000254 _H	DMATEST1 [R] XXXXXXXX XXXXX000 00000000 00000000				
000258 _H - 00027C _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000280 _H	SCR08 [R/W,W] 00000000	SMR08 [R/W,W] 00000000	SSR08 [R/W,R] 00001000	RDR08/TDR08 [R/W] 00000000	LIN-USART 8 with FIFO
000284 _H	ESCR08 [R/W] 00000X00	ECCR08 [R/W,R,W] 000000XX	FSR08 [R/W,R] XX00 0000	FCR08 [R/W] 0001 0000	
000288 _H	SCR09 [R/W,W] 00000000	SMR09 [R/W,W] 00000000	SSR09 [R/W,R] 00001000	RDR09/TDR09 [R/W] 00000000	LIN-USART 9 with FIFO
00028C _H	ESCR09 [R/W] 00000X00	ECCR09 [R/W,R,W] 000000XX	FSR09 [R/W,R] XX00 0000	FCR09 [R/W] 0001 0000	
000290 _H	SCR10 [R/W,W] 00000000	SMR10 [R/W,W] 00000000	SSR10 [R/W,R] 00001000	RDR10/TDR10 [R/W] 00000000	LIN-USART 10 with FIFO
000294 _H	ESCR10 [R/W] 00000X00	ECCR10 [R/W,R,W] 000000XX	FSR10 [R/W,R] XX00 0000	FCR10 [R/W] 0001 0000	
000298 _H	SCR11 [R/W,W] 00000000	SMR11 [R/W,W] 00000000	SSR11 [R/W,R] 00001000	RDR11/TDR11 [R/W] 00000000	LIN-USART 11 with FIFO
00029C _H	ESCR11 [R/W] 00000X00	ECCR11 [R/W,R,W] 000000XX	FSR11 [R/W,R] XX00 0000	FCR11 [R/W] 0001 0000	
0002A0 _H	SCR12 [R/W,W] 00000000	SMR12 [R/W,W] 00000000	SSR12 [R/W,R] 00001000	RDR12/TDR12 [R/W] 00000000	LIN-USART 12 with FIFO
0002A4 _H	ESCR12 [R/W] 00000X00	ECCR12 [R/W,R,W] 000000XX	FSR12 [R/W,R] XX00 0000	FCR12 [R/W] 0001 0000	
0002A8 _H	SCR13 [R/W,W] 00000000	SMR13 [R/W,W] 00000000	SSR13 [R/W,R] 00001000	RDR13/TDR13 [R/W] 00000000	LIN-USART 13 with FIFO
0002AC _H	ESCR13 [R/W] 00000X00	ECCR13 [R/W,R,W] 000000XX	FSR13 [R/W,R] XX00 0000	FCR13[R/W] 0001 0000	
0002B0 _H	SCR14 [R/W,W] 00000000	SMR14 [R/W,W] 00000000	SSR14 [R/W,R] 00001000	RDR14/TDR14 [R/W] 00000000	LIN-USART 14 with FIFO
0002B4 _H	ESCR14 [R/W] 00000X00	ECCR14 [R/W,R,W] 000000XX	FSR14 [R/W,R] XX00 0000	FCR14 [R/W] 0001 0000	

Address	Register				Block
	+0	+1	+2	+3	
0002B8 _H	SCR15 [R/W,W] 00000000	SMR15 [R/W,W] 00000000	SSR15 [R/W,R] 00001000	RDR15/TDR15 [R/W] 00000000	LIN-USART 15 with FIFO
0002BC _H	ESCR15 [R/W] 00000X00	ECCR15 [R/W,R,W] 000000XX	FSR15 [R/W,R] XX00 0000	FCR15 [R/W] 0001 0000	
0002C0 _H	BGR108 [R/W] 00000000	BGR008 [R/W] 00000000	BGR109 [R/W] 00000000	BGR009 [R/W] 00000000	Baudrate Generator LIN-USART 8-15
0002C4 _H	BGR110 [R/W] 00000000	BGR010 [R/W] 00000000	BGR111 [R/W] 00000000	BGR011 [R/W] 00000000	
0002C8 _H	BGR112 [R/W] 00000000	BGR012 [R/W] 00000000	BGR113 [R/W] 00000000	BGR013 [R/W] 00000000	
0002CC _H	BGR114 [R/W] 00000000	BGR014 [R/W] 00000000	BGR115 [R/W] 00000000	BGR015 [R/W] 00000000	
0002D0 _H	reserved	ICS45 [R/W] 00000000	reserved	ICS67 [R/W] 00000000	Input Capture 4-7
0002D4 _H	IPCP4 [R] XXXXXXXXXX XXXXXXXXX		IPCP5 [R] XXXXXXXXXX XXXXXXXXX		
0002D8 _H	IPCP6 [R] XXXXXXXXXX XXXXXXXXX		IPCP7 [R] XXXXXXXXXX XXXXXXXXX		
0002DC _H	OCS45 [R/W] - - -0 - -00 0000 - -00		OCS67 [R/W] - - -0 - -00 0000 - -00		Output Compare 4-7
0002E0 _H	OCCP4 [R/W] XXXXXXXXXX XXXXXXXXX		OCCP5 [R/W] XXXXXXXXXX XXXXXXXXX		
0002E4 _H	OCCP6 [R/W] XXXXXXXXXX XXXXXXXXX		OCCP7 [R/W] XXXXXXXXXX XXXXXXXXX		
0002E8 _H - 0002EC _H	reserved				
0002F0 _H	TCDT4 [R/W] XXXXXXXXXX XXXXXXXXX		reserved	TCCS4 [R/W] 00000000	Free Running Timer 4 (ICU 4-5)
0002F4 _H	TCDT5 [R/W] XXXXXXXXXX XXXXXXXXX		reserved	TCCS5 [R/W] 00000000	Free Running Timer 5 (ICU 6-7)
0002F8 _H	TCDT6 [R/W] XXXXXXXXXX XXXXXXXXX		reserved	TCCS6 [R/W] 00000000	Free Running Timer 6 (OCU 4-5)
0002FC _H	TCDT7 [R/W] XXXXXXXXXX XXXXXXXXX		reserved	TCCS7 [R/W] 00000000	Free Running Timer 7 (OCU 6-7)

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000300 _H	UDRC1 [W] 00000000	UDRC0 [W] 00000000	UDCR1 [R] 00000000	UDCR0 [R] 00000000	Up/Down Counter 0-1
000304 _H	UDCCH0 [R/W] 00000000	UDCCL0 [R/W] 00001000	reserved	UDCS0 [R/W] 00000000	
000308 _H	UDCCH1 [R/W] 00000000	UDCCL1 [R/W] 00001000	reserved	UDCS1 [R/W] 00000000	
00030C _H	reserved				
000310 _H	UDRC3 [W] 00000000	UDRC2 [W] 00000000	UDCR3 [R] 00000000	UDCR2 [R] 00000000	Up/Down Counter 2-3
000314 _H	UDCCH2 [R/W] 00000000	UDCCL2 [R/W] 00001000	reserved	UDCS2 [R/W] 00000000	
000318 _H	UDCCH3 [R/W] 00000000	UDCCL3 [R/W] 00001000	reserved	UDCS3 [R/W] 00000000	
00031C _H	reserved				
000320 _H	GCN13 [R/W] 00110010 00010000		reserved	GCN23 [R/W] - - - - 0000	PPG Control 12-15
000324 _H to 00032C _H	reserved				
000330 _H	PTMR12 [R] 11111111 11111111		PCSR12 [R/W] XXXXXXXXXX XXXXXXXXX		PPG 12
000334 _H	PDUT12 [R/W] XXXXXXXXXX XXXXXXXXX		PCNH12 [R/W] 0000000 -	PCNL12 [R/W] 000000 - 0	
000338 _H	PTMR13 [R] 11111111 11111111		PCSR13 [R/W] XXXXXXXXXX XXXXXXXXX		PPG 13
00033C _H	PDUT13 [R/W] XXXXXXXXXX XXXXXXXXX		PCNH13 [R/W] 0000000 -	PCNL13 [R/W] 000000 - 0	
000340 _H	PTMR14 [R] 11111111 11111111		PCSR14 [R/W] XXXXXXXXXX XXXXXXXXX		PPG 14
000344 _H	PDUT14 [R/W] XXXXXXXXXX XXXXXXXXX		PCNH14 [R/W] 0000000 -	PCNL14 [R/W] 000000 - 0	
000348 _H	PTMR15 [R] 11111111 11111111		PCSR15 [R/W] XXXXXXXXXX XXXXXXXXX		PPG 15
00034C _H	PDUT15 [R/W] XXXXXXXXXX XXXXXXXXX		PCNH15 [R/W] 0000000 -	PCNL15 [R/W] 000000 - 0	
000350 _H to 00035C _H	reserved				reserved
000360 _H	reserved	DACR [R/W] - - - - - 000	reserved	reserved	D/A Converter
000364 _H	DADR0 [R/W] - - - - - XX XXXXXXXXX		DADR1 [R/W] - - - - - XX XXXXXXXXX		

Address	Register				Block
	+0	+1	+2	+3	
000368 _H	IBCR2 [R/W] 00000000	IBSR2 [R] 00000000	ITBAH2 [R/W] ----- 00	ITBAL2 [R/W] 00000000	I2C 2
00036C _H	ITMKH2 [R/W] 00 ---- 11	ITMKL2 [R/W] 11111111	ISMK2 [R/W] 01111111	ISBA2 [R/W] - 00000000	
000370 _H	reserved	IDAR2 [R/W] 00000000	ICCR2 [R/W] 00011111	reserved	
000374 _H	IBCR3 [R/W] 00000000	IBSR3 [R] 00000000	ITBAH3 [R/W] ----- 00	ITBAL3 [R/W] 00000000	I2C 3
000378 _H	ITMKH3 [R/W] 00 ---- 11	ITMKL3 [R/W] 11111111	ISMK3 [R/W] 01111111	ISBA3 [R/W] - 00000000	
00037C _H	reserved	IDAR3 [R/W] 00000000	ICCR3 [R/W] 00011111	reserved	
000380 _H	IBCR4 [R/W] 00000000	IBSR4 [R] 00000000	ITBAH4 [R/W] ----- 00	ITBAL4 [R/W] 00000000	I2C 4
000384 _H	ITMKH4 [R/W] 00 ---- 11	ITMKL4 [R/W] 11111111	ISMK4 [R/W] 01111111	ISBA4 [R/W] - 00000000	
000388 _H	reserved	IDAR4 [R/W] 00000000	ICCR4 [R/W] 00011111	reserved	
00038C _H	reserved				
000390 _H	ROMS [R,R/W] ² 11111000 00000000		reserved		ROM Select register
000394 _H	IBCR5 [R/W] 00000000	IBSR5 [R] 00000000	ITBAH5 [R/W] ----- 00	ITBAL5 [R/W] 00000000	I2C 5
000398 _H	ITMKH5 [R/W] 00 ---- 11	ITMKL5 [R/W] 11111111	ISMK5 [R/W] 01111111	ISBA5 [R/W] - 00000000	
00039C _H	reserved	IDAR5 [R/W] 00000000	ICCR5 [R/W] 00011111	reserved	
0003A0 _H	IBCR6 [R/W] 00000000	IBSR6 [R] 00000000	ITBAH6 [R/W] ----- 00	ITBAL6 [R/W] 00000000	I2C 6
0003A4 _H	ITMKH6 [R/W] 00 ---- 11	ITMKL6 [R/W] 11111111	ISMK6 [R/W] 01111111	ISBA6 [R/W] - 00000000	
0003A8 _H	reserved	IDAR6 [R/W] 00000000	ICCR6 [R/W] 00011111	reserved	
0003AC _H	IBCR7 [R/W] 00000000	IBSR7 [R] 00000000	ITBAH7 [R/W] ----- 00	ITBAL7 [R/W] 00000000	I2C 7
0003B0 _H	ITMKH7 [R/W] 00 ---- 11	ITMKL7 [R/W] 11111111	ISMK7 [R/W] 01111111	ISBA7 [R/W] - 00000000	
0003B4 _H	reserved	IDAR7 [R/W] 00000000	ICCR7 [R/W] 00011111	reserved	
0003B8 _H	reserved				
0003BC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0003C0 _H	reserved				
0003C4 _H	reserved			ISIZE [R/W] ----- 10	I-Cache control
0003C8 _H - 0003E0 _H	reserved				
0003E4 _H	reserved			ICHCR [R/W] 0 - 000000	I-Cache control
0003E8 _H - 0003EC _H	reserved				
0003F0 _H	BSD0 [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Bit Search Module
0003F4 _H	BSD1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003F8 _H	BSDC [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003FC _H	BSRR [R] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000400 _H - 00043C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
000440 _H	ICR00 [R/W] --- 11111	ICR01 [R/W] --- 11111	ICR02 [R/W] --- 11111	ICR03 [R/W] --- 11111	Interrupt Control register
000444 _H	ICR04 [R/W] --- 11111	ICR05 [R/W] --- 11111	ICR06 [R/W] --- 11111	ICR07 [R/W] --- 11111	
000448 _H	ICR08 [R/W] --- 11111	ICR09 [R/W] --- 11111	ICR10 [R/W] --- 11111	ICR11 [R/W] --- 11111	
00044C _H	ICR12 [R/W] --- 11111	ICR13 [R/W] --- 11111	ICR14 [R/W] --- 11111	ICR15 [R/W] --- 11111	
000450 _H	ICR16 [R/W] --- 11111	ICR17 [R/W] --- 11111	ICR18 [R/W] --- 11111	ICR19 [R/W] --- 11111	
000454 _H	ICR20 [R/W] --- 11111	ICR21 [R/W] --- 11111	ICR22 [R/W] --- 11111	ICR23 [R/W] --- 11111	
000458 _H	ICR24 [R/W] --- 11111	ICR25 [R/W] --- 11111	ICR26 [R/W] --- 11111	ICR27 [R/W] --- 11111	
00045C _H	ICR28 [R/W] --- 11111	ICR29 [R/W] --- 11111	ICR30 [R/W] --- 11111	ICR31 [R/W] --- 11111	
000460 _H	ICR32 [R/W] --- 11111	ICR33 [R/W] --- 11111	ICR34 [R/W] --- 11111	ICR35 [R/W] --- 11111	
000464 _H	ICR36 [R/W] --- 11111	ICR37 [R/W] --- 11111	ICR38 [R/W] --- 11111	ICR39 [R/W] --- 11111	
000468 _H	ICR40 [R/W] --- 11111	ICR41 [R/W] --- 11111	ICR42 [R/W] --- 11111	ICR43 [R/W] --- 11111	
00046C _H	ICR44 [R/W] --- 11111	ICR45 [R/W] --- 11111	ICR46 [R/W] --- 11111	ICR47 [R/W] --- 11111	
000470 _H	ICR48 [R/W] --- 11111	ICR49 [R/W] --- 11111	ICR50 [R/W] --- 11111	ICR51 [R/W] --- 11111	
000474 _H	ICR52 [R/W] --- 11111	ICR53 [R/W] --- 11111	ICR54 [R/W] --- 11111	ICR55 [R/W] --- 11111	
000478 _H	ICR56 [R/W] --- 11111	ICR57 [R/W] --- 11111	ICR58 [R/W] --- 11111	ICR59 [R/W] --- 11111	
00047C _H	ICR60 [R/W] --- 11111	ICR61 [R/W] --- 11111	ICR62 [R/W] --- 11111	ICR63 [R/W] --- 11111	
000480 _H	RSRR [R/W] 10000000	STCR [R/W] 00110011	TBCR [R/W] 00XXX – 00	CTBR [W] XXXXXXXXXX	Clock Control Unit
000484 _H	CLKR [R/W] ---- 0000	WPR [R/W] XXXX XXX0	DIVR0 [R/W] 00000011	DIVR1 [R/W] 00000000	
000488 _H	CTEST [R/W] XXXX00XX	reserved	reserved	reserved	C-Unit Test (hidden)
00048C _H	PLLDIVM [R/W] ---- 0000	PLLDIVN [R/W] -- 000000	PLLDIVG [R/W] ---- 0000	PLLMULG [W] 00000000	PLL Clock Gear Unit
000490 _H	PLLCTRL [R/W] ---- 0000	reserved	reserved	reserved	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000494 _H	OSCC1 [R/W] ----- 010	reserved	OSCC2 [R/W] ----- 010	reserved	Main/Sub Oscillator Control
000498 _H	PORTEN [R/W] ----- 000	CMTST [R/W] ³ 0000 0000	PPMUX [R/W] ⁴ 00000000 00000000		Port Input Enable Control / PortMux Control
00049C _H	PPMUX2 [R/W] ⁵ -- 00 0000 -----		reserved	reserved	PortMux Control
0004A0 _H	reserved	WTCER [R/W] ----- 00	WTCR [R/W] 00000000 000 – 00 – 0		Watchdog Timer (Real Time Clock)
0004A4 _H	reserved	WTBR [R/W] --- XXXXX XXXXXXXX XXXXXXXX			
0004A8 _H	WTHR [R/W] --- 00000	WTMR [R/W] -- 000000	WTSR [R/W] -- 000000	reserved	
0004AC _H	CSVTR [R/W] --- 00010	CSVCR [R/W] 00011100	CSCFG [R/W] 0X000000	CMCFG [R/W] 00000000	Clock- Supervisor / Se- lector / Monitor
0004B0 _H	CUCR [R/W] ----- 0 -- 00		CUTD [R/W] 10000000 00000000		Calibration Unit of Sub Oscillation
0004B4 _H	CUTR1 [R] ----- 00000000		CUTR2 [R] 00000000 00000000		
0004B8 _H	CMPR [R/W] -- 000010 11111101		reserved	CMCR [R/W] - 001 -- 00	
0004BC _H	CMT1 [R/W] 00000000 1 --- 0000		CMT2 [R/W] -- 000000 -- 000000		
0004C0 _H	CANPRE [R/W] -- 00 0000	CANCKD [R/W] -- 000000 ⁶			CAN Clock
0004C4 _H	LVSEL [R/W] 00000111	LVDET [R/W] 0000 0 - 00	HWWDE [R/W] --- 0 -- 00	HWWD [R/W,W] 00011000	LV Detection / Hardware- Watchdog
0004C8 _H	OSCRH [R/W] 000 -- 001	OSCRL [R/W] ----- 000	WPCRH [R/W] 000 -- 000	WPCRL [R/W] ----- 00	Main-/Sub-Oscil- lation Stabilisa- tion Timer
0004CC _H	OSCCR [R/W] ----- 00	reserved	REGSEL [R/W] -- 000110	REGCTR [R/W] --- X -- 00	Main- Oscillation Standby Control Main/Sub Regula- tor Control
0004D0 _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
0004D4 _H	SHDE [R/W] 0 - - - - - 0	reserved	EXTE [R/W] 0000 0000	EXTF [R/W0] 0000 0000	Shutdown Control
0004D8 _H	EXTLV [R/W] 0000 0000 0000 0000		reserved	SHDINT [R/W] - - - - 0000	
0004DC _H	PLL2DIVM [R/W] - - - - 0000	PLL2DIVN [R/W] - - 000000	PLL2DIVG [R/W] - - - - 0000	PLL2MULG [R/W] 00000000	PLL2 Clock Control (FlexRay, USB)
0004E0 _H	PLL2CTRL [R/W] - - - - 0000	reserved	CLKR2 [R/W] 000 00000	reserved	
0004E4 _H	PLL3DIVM [R/W] - - - - 0000	PLL3DIVN [R/W] - - 000000	PLL3DIVG [R/W] - - - - 0000	PLL3MULG [R/W] 00000000	PLL3 Clock Control (USB)
0004E8 _H	PLL3CTRL [R/W] - - - - 0000	reserved	MLBCNT [R/W] 000 - - - - 0	MLBPRES [R/W] - - 000000	MediaLB Clock Control
0004EC _H to 0004F0 _H	reserved				reserved
0004F4 _H	MPLLDIVM [R/W] - - - - 0000	MPLLDIVN [R/W] - - 00 0000	MPLLDIVG [R/W] - - - - 0000	MPLLMULG [R/W] 0000 0000	MediaLB PLL Clock Gear Unit
0004F8 _H	MPLLCTRL [R/W] - - - - 0000	reserved			
0004FC _H	reserved				
000500 _H	GCN14 [R/W] 00110010 00010000		reserved	GCN24 [R/W] - - - - 0000	PPG Control 16-19
000504 _H	GCN15 [R/W] 00110010 00010000		reserved	GCN25 [R/W] - - - - 0000	PPG Control 20-23
000508 _H	GCN16 [R/W] 00110010 00010000		reserved	GCN26 [R/W] - - - - 0000	PPG Control 24-27
00050C _H	GCN17 [R/W] 00110010 00010000		reserved	GCN27 [R/W] - - - - 0000	PPG Control 28-31
000510 _H	PTMR16 [R] 11111111 11111111		PCSR16 [R/W] XXXXXXXX XXXXXXXX		PPG 16
000514 _H	PDUT16 [R/W] XXXXXXXX XXXXXXXX		PCNH16 [R/W] 0000000 -	PCNL16 [R/W] 000000 - 0	
000518 _H	PTMR17 [R] 11111111 11111111		PCSR17 [R/W] XXXXXXXX XXXXXXXX		PPG 17
00051C _H	PDUT17 [R/W] XXXXXXXX XXXXXXXX		PCNH17 [R/W] 0000000 -	PCNL17 [R/W] 000000 - 0	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000520 _H	PTMR18 [R] 11111111 11111111		PCSR18 [R/W] XXXXXXXX XXXXXXXX		PPG 18
000524 _H	PDUT18 [R/W] XXXXXXXX XXXXXXXX		PCNH18 [R/W] 0000000 -	PCNL18 [R/W] 000000 - 0	
000528 _H	PTMR19 [R] 11111111 11111111		PCSR19 [R/W] XXXXXXXX XXXXXXXX		PPG 19
00052C _H	PDUT19 [R/W] XXXXXXXX XXXXXXXX		PCNH19 [R/W] 0000000 -	PCNL19 [R/W] 000000 - 0	
000530 _H	PTMR20 [R] 11111111 11111111		PCSR20 [R/W] XXXXXXXX XXXXXXXX		PPG 20
000534 _H	PDUT20 [R/W] XXXXXXXX XXXXXXXX		PCNH20 [R/W] 0000000 -	PCNL20 [R/W] 000000 - 0	
000538 _H	PTMR21 [R] 11111111 11111111		PCSR21 [R/W] XXXXXXXX XXXXXXXX		PPG 21
00053C _H	PDUT21 [R/W] XXXXXXXX XXXXXXXX		PCNH21 [R/W] 0000000 -	PCNL21 [R/W] 000000 - 0	
000540 _H	PTMR22 [R] 11111111 11111111		PCSR22 [R/W] XXXXXXXX XXXXXXXX		PPG 22
000544 _H	PDUT22 [R/W] XXXXXXXX XXXXXXXX		PCNH22 [R/W] 0000000 -	PCNL22 [R/W] 000000 - 0	
000548 _H	PTMR23 [R] 11111111 11111111		PCSR23 [R/W] XXXXXXXX XXXXXXXX		PPG 23
00054C _H	PDUT23 [R/W] XXXXXXXX XXXXXXXX		PCNH23 [R/W] 0000000 -	PCNL23 [R/W] 000000 - 0	
000550 _H	PTMR24 [R] 11111111 11111111		PCSR24 [R/W] XXXXXXXX XXXXXXXX		PPG 24
000554 _H	PDUT24 [R/W] XXXXXXXX XXXXXXXX		PCNH24 [R/W] 0000000 -	PCNL24 [R/W] 000000 - 0	
000558 _H	PTMR25 [R] 11111111 11111111		PCSR25 [R/W] XXXXXXXX XXXXXXXX		PPG 25
00055C _H	PDUT25 [R/W] XXXXXXXX XXXXXXXX		PCNH25 [R/W] 0000000 -	PCNL25 [R/W] 000000 - 0	
000560 _H	PTMR26 [R] 11111111 11111111		PCSR26 [R/W] XXXXXXXX XXXXXXXX		PPG 26
000564 _H	PDUT26 [R/W] XXXXXXXX XXXXXXXX		PCNH26 [R/W] 0000000 -	PCNL26 [R/W] 000000 - 0	
000568 _H	PTMR27 [R] 11111111 11111111		PCSR27 [R/W] XXXXXXXX XXXXXXXX		PPG 27
00056C _H	PDUT27 [R/W] XXXXXXXX XXXXXXXX		PCNH27 [R/W] 0000000 -	PCNL27 [R/W] 000000 - 0	

Address	Register				Block
	+0	+1	+2	+3	
000570 _H	PTMR28 [R] 11111111 11111111		PCSR28 [R/W] XXXXXXXX XXXXXXXX		PPG 28
000574 _H	PDUT28 [R/W] XXXXXXXX XXXXXXXX		PCNH28 [R/W] 0000000 -	PCNL28 [R/W] 000000 - 0	
000578 _H	PTMR29 [R] 11111111 11111111		PCSR29 [R/W] XXXXXXXX XXXXXXXX		PPG 29
00057C _H	PDUT29 [R/W] XXXXXXXX XXXXXXXX		PCNH29 [R/W] 0000000 -	PCNL29 [R/W] 000000 - 0	
000580 _H	PTMR30 [R] 11111111 11111111		PCSR30 [R/W] XXXXXXXX XXXXXXXX		PPG 30
000584 _H	PDUT30 [R/W] XXXXXXXX XXXXXXXX		PCNH30 [R/W] 0000000 -	PCNL30 [R/W] 000000 - 0	
000588 _H	PTMR31 [R] 11111111 11111111		PCSR31 [R/W] XXXXXXXX XXXXXXXX		PPG 31
00058C _H	PDUT31 [R/W] XXXXXXXX XXXXXXXX		PCNH31 [R/W] 0000000 -	PCNL31 [R/W] 000000 - 0	
000590 _H	TMRLR8 [W] XXXXXXXX XXXXXXXX		TMR8 [R] XXXXXXXX XXXXXXXX		Reload Timer 8 (PPG 16-19)
000594 _H	reserved		TMCSRH8 [R/W] - - - 00000	TMCSRL8 [R/W] 0 - 000000	
000598 _H	TMRLR9 [W] XXXXXXXX XXXXXXXX		TMR9 [R] XXXXXXXX XXXXXXXX		Reload Timer 9 (PPG 16-19)
00059C _H	reserved		TMCSRH9 [R/W] - - - 00000	TMCSRL9 [R/W] 0 - 000000	
0005A0 _H	TMRLR10 [W] XXXXXXXX XXXXXXXX		TMR10 [R] XXXXXXXX XXXXXXXX		Reload Timer 10 (PPG 20-23)
0005A4 _H	reserved		TMCSRH10 [R/W] - - - 00000	TMCSRL10 [R/W] 0 - 000000	
0005A8 _H	TMRLR11 [W] XXXXXXXX XXXXXXXX		TMR11 [R] XXXXXXXX XXXXXXXX		Reload Timer 11 (PPG 20-23)
0005AC _H	reserved		TMCSRH11 [R/W] - - - 00000	TMCSRL11 [R/W] 0 - 000000	
0005B0 _H	TMRLR12 [W] XXXXXXXX XXXXXXXX		TMR12 [R] XXXXXXXX XXXXXXXX		Reload Timer 12 (PPG 24-27)
0005B4 _H	reserved		TMCSRH12 [R/W] - - - 00000	TMCSRL12 [R/W] 0 - 000000	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0005B8 _H	TMRLR13 [W] XXXXXXXXXX XXXXXXXXXX		TMR13 [R] XXXXXXXXXX XXXXXXXXXX		Reload Timer 13 (PPG 24-27)
0005BC _H	reserved		TMCSRH13 [R/W] - - - 00000	TMCSRL13 [R/W] 0 - 000000	
0005C0 _H	TMRLR14 [W] XXXXXXXXXX XXXXXXXXXX		TMR14 [R] XXXXXXXXXX XXXXXXXXXX		Reload Timer 14 (PPG 28-31)
0005C4 _H	reserved		TMCSRH14 [R/W] - - - 00000	TMCSRL14 [R/W] 0 - 000000	
0005C8 _H	TMRLR15 [W] XXXXXXXXXX XXXXXXXXXX		TMR15 [R] XXXXXXXXXX XXXXXXXXXX		Reload Timer 15 (PPG 28-31)
0005CC _H	reserved		TMCSRH15 [R/W] - - - 00000	TMCSRL15 [R/W] 0 - 000000	
0005D0 _H	TMR89 [R] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				Reload Timer 8 + 9
0005D4 _H	TMR1011 [R] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				Reload Timer 10 + 11
0005D8 _H	TMR1213 [R] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				Reload Timer 12 + 13
0005DC _H	TMR1415 [R] XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX				Reload Timer 14 + 15
0005E0 _H	AD1ERH [R/W] 00000000 00000000		AD1ERL [R/W] 00000000 00000000		A/D Converter 1
0005E4 _H	AD1CS1 [R/W] 00000000	AD1CS0 [R/W] 00000000	AD1CR1 [R] 000000XX	AD1CR0 [R] XXXXXXXXXX	
0005E8 _H	AD1CT1 [R/W] 00010000	AD1CT0 [R/W] 00101100	AD1SCH [R/W] - - - 00000	AD1ECH [R/W] - - - 00000	
0005EC _H	reserved				
0005F0 _H	reserved	ICS89 [R/W] 00000000	reserved	reserved	Input Capture 8-9
0005F4 _H	IPCP8 [R] XXXXXXXXXX XXXXXXXXXX		IPCP9 [R] XXXXXXXXXX XXXXXXXXXX		
0005F8 _H	reserved				
0005FC _H	reserved				
000600 _H					
000604 _H					
000608 _H	TCDT8 [R/W] XXXXXXXXXX XXXXXXXXXX		reserved	TCCS8 [R/W] 00000000	Free Running Timer 8 (ICU 8-9)

Address	Register				Block
	+0	+1	+2	+3	
00060C _H	TCDT9 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS9 [R/W] 00000000	Free Running Timer 9
000610 _H	TCDT10 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS10 [R/W] 00000000	Free Running Timer 10
000614 _H	TCDT11 [R/W] XXXXXXXXX XXXXXXXXX		reserved	TCCS11 [R/W] 00000000	Free Running Timer 11
000618 _H - 00063C _H	reserved				
000640 _H	ASR0 [R/W] 00000000 00000000		ACR0 [R/W] 1111**00 00100000 ⁷		External Bus Unit
000644 _H	ASR1 [R/W] XXXXXXXXX XXXXXXXXX		ACR1 [R/W] XXXXXXXXX XXXXXXXXX		
000648 _H	ASR2 [R/W] XXXXXXXXX XXXXXXXXX		ACR2 [R/W] XXXXXXXXX XXXXXXXXX		
00064C _H	ASR3 [R/W] XXXXXXXXX XXXXXXXXX		ACR3 [R/W] XXXXXXXXX XXXXXXXXX		
000650 _H	ASR4 [R/W] XXXXXXXXX XXXXXXXXX		ACR4 [R/W] XXXXXXXXX XXXXXXXXX		
000654 _H	ASR5 [R/W] XXXXXXXXX XXXXXXXXX		ACR5 [R/W] XXXXXXXXX XXXXXXXXX		
000658 _H	ASR6 [R/W] XXXXXXXXX XXXXXXXXX		ACR6 [R/W] XXXXXXXXX XXXXXXXXX		
00065C _H	ASR7 [R/W] XXXXXXXXX XXXXXXXXX		ACR7 [R/W] XXXXXXXXX XXXXXXXXX		
000660 _H	AWR0 [R/W] 01111111 11111011		AWR1 [R/W] XXXXXXXXX XXXXXXXXX		
000664 _H	AWR2 [R/W] XXXXXXXXX XXXXXXXXX		AWR3 [R/W] XXXXXXXXX XXXXXXXXX		
000668 _H	AWR4 [R/W] XXXXXXXXX XXXXXXXXX		AWR5 [R/W] XXXXXXXXX XXXXXXXXX		
00066C _H	AWR6 [R/W] XXXXXXXXX XXXXXXXXX		AWR7 [R/W] XXXXXXXXX XXXXXXXXX		

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000670 _H	MCRA [R/W] XXXXXXXX	MCRB [R/W] XXXXXXXX	reserved		External Bus Unit
000674 _H	reserved				
000678 _H	IOWR0 [R/W] XXXXXXXX	IOWR1 [R/W] XXXXXXXX	IOWR2 [R/W] XXXXXXXX	IOWR3 [R/W] XXXXXXXX	
00067C _H	reserved				
000680 _H	CSER [R/W] 00000001	CHER [R/W] 11111111	reserved	TCR [R/W] 0000****8	
000684 _H	RCRH [R/W] 00XXXXXX	RCRL [R/W] XXXX0XXX	reserved	reserved	
000688 _H	RCO0H0 [R/W] 11111111	RCO0L0 [R/W] 0000 0000	RCO0H1 [R/W] 11111111	RCO0L1 [R/W] 0000 0000	A/D Converter 0 Range Compara- tor
00068C _H	RCO0H2 [R/W] 11111111	RCO0L2 [R/W] 0000 0000	RCO0H3 [R/W] 11111111	RCO0L3 [R/W] 0000 0000	
000690 _H	RCO0IRS [R/W] 00000000 00000000 00000000 00000000				
000694 _H	RCO0OF [R] 00000000 00000000 00000000 00000000				
000698 _H	RCO0INT [R/W0] 00000000 00000000 00000000 00000000				
00069C _H	reserved				
0006A0 _H	AD0CC0 [R/W] 0000 0000	AD0CC1 [R/W] 0000 0000	AD0CC2 [R/W] 0000 0000	AD0CC3 [R/W] 0000 0000	A/D Converter 0 Channel Control
0006A4 _H	AD0CC4 [R/W] 0000 0000	AD0CC5 [R/W] 0000 0000	AD0CC6 [R/W] 0000 0000	AD0CC7 [R/W] 0000 0000	
0006A8 _H	AD0CC8 [R/W] 0000 0000	AD0CC9 [R/W] 0000 0000	AD0CC10 [R/W] 0000 0000	AD0CC11 [R/W] 0000 0000	
0006AC _H	AD0CC12 [R/W] 0000 0000	AD0CC13 [R/W] 0000 0000	AD0CC14 [R/W] 0000 0000	AD0CC15 [R/W] 0000 0000	
0006B0 _H	AD0CS2 [RW] 0000 - - 00	reserved			A/D Converter 0 Control register 2

Address	Register				Block
	+0	+1	+2	+3	
0006B4 _H	RCO1H0 [R/W] 11111111	RCO1L0 [R/W] 0000 0000	RCO1H1 [R/W] 11111111	RCO1L1 [R/W] 0000 0000	A/D Converter 1 Range Compara- tor
0006B8 _H	RCO1H2 [R/W] 11111111	RCO1L2 [R/W] 0000 0000	RCO1H3 [R/W] 11111111	RCO1L3 [R/W] 0000 0000	
0006BC _H	RCO1IRS [R/W] 00000000 00000000 00000000 00000000				
0006C0 _H	RCO1OF [R] 00000000 00000000 00000000 00000000				
0006C4 _H	RCO1INT [R/W0] 00000000 00000000 00000000 00000000				
0006C8 _H	reserved				A/D Converter 1 Channel Control
0006CC _H	AD1CC0 [R/W] 0000 0000	AD1CC1 [R/W] 0000 0000	AD1CC2 [R/W] 0000 0000	AD1CC3 [R/W] 0000 0000	
0006D0 _H	AD1CC4 [R/W] 0000 0000	AD1CC5 [R/W] 0000 0000	AD1CC6 [R/W] 0000 0000	AD1CC7 [R/W] 0000 0000	
0006D4 _H	AD1CC8 [R/W] 0000 0000	AD1CC9 [R/W] 0000 0000	AD1CC10 [R/W] 0000 0000	AD1CC11 [R/W] 0000 0000	
0006D8 _H	AD1CC12 [R/W] 0000 0000	AD1CC13 [R/W] 0000 0000	AD1CC14 [R/W] 0000 0000	AD1CC15 [R/W] 0000 0000	
0006DC _H	AD1CS2 [RW] 0000 - - 00	reserved			A/D Converter 1 Control register 2
0006E0 _H	ADC0D0 [R] ----- XX XXXXXXXXX		ADC0D1 [R] ----- XX XXXXXXXXX		A/D Converter 0 Channel Data registers
0006E4 _H	ADC0D2 [R] ----- XX XXXXXXXXX		ADC0D3 [R] ----- XX XXXXXXXXX		
0006E8 _H	ADC0D4 [R] ----- XX XXXXXXXXX		ADC0D5 [R] ----- XX XXXXXXXXX		
0006EC _H	ADC0D6 [R] ----- XX XXXXXXXXX		ADC0D7 [R] ----- XX XXXXXXXXX		
0006F0 _H	ADC0D8 [R] ----- XX XXXXXXXXX		ADC0D9 [R] ----- XX XXXXXXXXX		
0006F4 _H	ADC0D10 [R] ----- XX XXXXXXXXX		ADC0D11 [R] ----- XX XXXXXXXXX		
0006F8 _H	ADC0D12 [R] ----- XX XXXXXXXXX		ADC0D13 [R] ----- XX XXXXXXXXX		
0006FC _H	ADC0D14 [R] ----- XX XXXXXXXXX		ADC0D015 [R] ----- XX XXXXXXXXX		
000700 _H	ADC0D16 [R] ----- XX XXXXXXXXX		ADC0D17 [R] ----- XX XXXXXXXXX		

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000704 _H	ADC0D18 [R] ----- XX XXXXXXXXX		ADC0D19 [R] ----- XX XXXXXXXXX		A/D Converter 0 Channel Data registers
000708 _H	ADC0D20 [R] ----- XX XXXXXXXXX		ADC0D21 [R] ----- XX XXXXXXXXX		
00070C _H	ADC0D22 [R] ----- XX XXXXXXXXX		ADC0D23 [R] ----- XX XXXXXXXXX		
000710 _H	ADC0D24 [R] ----- XX XXXXXXXXX		ADC0D25 [R] ----- XX XXXXXXXXX		
000714 _H	ADC0D26 [R] ----- XX XXXXXXXXX		ADC0D27 [R] ----- XX XXXXXXXXX		
000718 _H	ADC0D28 [R] ----- XX XXXXXXXXX		ADC0D29 [R] ----- XX XXXXXXXXX		
00071C _H	ADC0D30 [R] ----- XX XXXXXXXXX		ADC0D31 [R] ----- XX XXXXXXXXX		
000720 _H	ADC1D0 [R] ----- XX XXXXXXXXX		ADC1D1 [R] ----- XX XXXXXXXXX		A/D Converter 1 Channel Data registers
000724 _H	ADC1D2 [R] ----- XX XXXXXXXXX		ADC1D3 [R] ----- XX XXXXXXXXX		
000728 _H	ADC1D4 [R] ----- XX XXXXXXXXX		ADC1D5 [R] ----- XX XXXXXXXXX		
00072C _H	ADC1D6 [R] ----- XX XXXXXXXXX		ADC1D7 [R] ----- XX XXXXXXXXX		
000730 _H	ADC1D8 [R] ----- XX XXXXXXXXX		ADC1D9 [R] ----- XX XXXXXXXXX		
000734 _H	ADC1D10 [R] ----- XX XXXXXXXXX		ADC1D11 [R] ----- XX XXXXXXXXX		
000738 _H	ADC1D12 [R] ----- XX XXXXXXXXX		ADC1D13 [R] ----- XX XXXXXXXXX		
00073C _H	ADC1D14 [R] ----- XX XXXXXXXXX		ADC1D015 [R] ----- XX XXXXXXXXX		
000740 _H	ADC1D16 [R] ----- XX XXXXXXXXX		ADC1D17 [R] ----- XX XXXXXXXXX		
000744 _H	ADC1D18 [R] ----- XX XXXXXXXXX		ADC1D19 [R] ----- XX XXXXXXXXX		
000748 _H	ADC1D20 [R] ----- XX XXXXXXXXX		ADC1D21 [R] ----- XX XXXXXXXXX		
00074C _H	ADC1D22 [R] ----- XX XXXXXXXXX		ADC1D23 [R] ----- XX XXXXXXXXX		

Address	Register				Block
	+0	+1	+2	+3	
000750 _H	ADC1D24 [R] ----- XX XXXXXXXXX		ADC1D25 [R] ----- XX XXXXXXXXX		A/D Converter 1 Channel Data registers
000754 _H	ADC1D26 [R] ----- XX XXXXXXXXX		ADC1D27 [R] ----- XX XXXXXXXXX		
000758 _H	ADC1D28 [R] ----- XX XXXXXXXXX		ADC1D29 [R] ----- XX XXXXXXXXX		
00075C _H	ADC1D30 [R] ----- XX XXXXXXXXX		ADC1D31 [R] ----- XX XXXXXXXXX		
000760 _H - 0007F8 _H	reserved				
0007FC _H	reserved	MODR [W] XXXXXXXX	reserved	reserved	Mode Register
000800 _H - 000BFC _H	reserved				
000C00 _H	reserved			IOS [R/W] -----10 ^{*9}	I-Unit
000C04 _H	EIRR2 [R/W] XXXXXXXX	ENIR2 [R/W] 00000000	ELVR2 [R/W] 00000000 00000000		Ext. INT 16-23
000C08 _H	EIRR3 [R/W] XXXXXXXX	ENIR3 [R/W] 00000000	ELVR3 [R/W] 00000000 00000000		Ext. INT 24-31
000C0C _H - 000CFC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000D00 _H	PDRD00 [R] XXXXXXXX	PDRD01 [R] XXXXXXXX	PDRD02 [R] XXXXXXXX	PDRD03 [R] XXXXXXXX	General IO Port Direct Read Data register
000D04 _H	PDRD04 [R] XXXXXXXX	PDRD05 [R] XXXXXXXX	PDRD06 [R] XXXXXXXX	PDRD07 [R] XXXXXXXX	
000D08 _H	PDRD08 [R] XXXXXXXX	PDRD09 [R] XXXXXXXX	PDRD10 [R] XXXXXXXX	PDRD11 [R] XXXXXXXX	
000D0C _H	PDRD12 [R] XXXXXXXX	PDRD13 [R] XXXXXXXX	PDRD14 [R] XXXXXXXX	PDRD15 [R] XXXXXXXX	
000D10 _H	PDRD16 [R] XXXXXXXX	PDRD17 [R] XXXXXXXX	PDRD18 [R] XXXXXXXX	PDRD19 [R] XXXXXXXX	
000D14 _H	PDRD20 [R] XXXXXXXX	PDRD21 [R] XXXXXXXX	PDRD22 [R] XXXXXXXX	PDRD23 [R] XXXXXXXX	
000D18 _H	PDRD24 [R] XXXXXXXX	PDRD25 [R] XXXXXXXX	PDRD26 [R] XXXXXXXX	PDRD27 [R] XXXXXXXX	
000D1C _H	PDRD28 [R] XXXXXXXX	PDRD29 [R] XXXXXXXX	PDRD30 [R] XXXXXXXX	PDRD31 [R] XXXXXXXX	
000D20 _H	PDRD32 [R] XXXXXXXX	PDRD33 [R] XXXXXXXX	PDRD34 [R] XXXXXXXX	PDRD35 [R] XXXXXXXX	
000D24 _H	PDRD36 [R] XXXXXXXX	PDRD37 [R] XXXXXXXX	PDRD38 [R] XXXXXXXX	PDRD39 [R] XXXXXXXX	
000D28 _H	PDRD40 [R] XXXXXXXX	reserved	reserved	reserved	
000D2C _H - 000D3C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
000D40 _H	DDR00 [R/W] 00000000	DDR01 [R/W] 00000000	DDR02 [R/W] 00000000	DDR03 [R/W] 00000000	General IO Port Data Direction register
000D44 _H	DDR04 [R/W] 00000000	DDR05 [R/W] 00000000	DDR06 [R/W] 00000000	DDR07 [R/W] 00000000	
000D48 _H	DDR08 [R/W] 00000000	DDR09 [R/W] 00000000	DDR10 [R/W] 00000000	DDR11 [R/W] 00000000	
000D4C _H	DDR12 [R/W] 00000000	DDR13 [R/W] 00000000	DDR14 [R/W] 00000000	DDR15 [R/W] 00000000	
000D50 _H	DDR16 [R/W] 00000000	DDR17 [R/W] 00000000	DDR18 [R/W] 00000000	DDR19 [R/W] 00000000	
000D54 _H	DDR20 [R/W] 00000000	DDR21 [R/W] 00000000	DDR22 [R/W] 00000000	DDR23 [R/W] 00000000	
000D58 _H	DDR24 [R/W] 00000000	DDR25 [R/W] 00000000	DDR26 [R/W] 00000000	DDR27 [R/W] 00000000	
000D5C _H	DDR28 [R/W] 00000000	DDR29 [R/W] 00000000	DDR30 [R/W] 00000000	DDR31 [R/W] 00000000	
000D60 _H	DDR32 [R/W] 00000000	DDR33 [R/W] 00000000	DDR34 [R/W] 00000000	DDR35 [R/W] 00000000	
000D64 _H	DDR36 [R/W] 00000000	DDR37 [R/W] 00000000	DDR38 [R/W] 00000000	DDR39 [R/W] 00000000	
000D68 _H	DDR40 [R/W] 00000000	reserved	reserved	reserved	
000D6C _H - 000D7C _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
	Note ###: The initial values of PFR00 to PFR10 (for external bus interface) on MB91FV460B depend on the series mode. For old devices, the PFRs are initial 0xFF always. For new devices the PFRs are initial 0x00 in internal vector fetch mode (MD[2:0]=000), and 0xFF in external vector fetch mode (MD[2:0]=001).				Port Function register
000D80 _H	PFR00 [R/W] #####	PFR01 [R/W] #####	PFR02 [R/W] #####	PFR03 [R/W] #####	
000D84 _H	PFR04 [R/W] #####	PFR05 [R/W] #####	PFR06 [R/W] #####	PFR07 [R/W] #####	
000D88 _H	PFR08 [R/W] #####	PFR09 [R/W] #####	PFR10 [R/W] - #0#####	PFR11 [R/W] - - - - - 00	
000D8C _H	PFR12 [R/W] 00000000	PFR13 [R/W] 00000000	PFR14 [R/W] 00000000	PFR15 [R/W] 00000000	
000D90 _H	PFR16 [R/W] 00000000	PFR17 [R/W] 00000000	PFR18 [R/W] - 000 - 000	PFR19 [R/W] 00000000	
000D94 _H	PFR20 [R/W] - 000 - 000	PFR21 [R/W] - 000 - 000	PFR22 [R/W] 00000000	PFR23 [R/W] 00000000	
000D98 _H	PFR24 [R/W] 00000000	PFR25 [R/W] 00000000	PFR26 [R/W] 00000000	PFR27 [R/W] 00000000	
000D9C _H	PFR28 [R/W] 00000000	PFR29 [R/W] 00000000	PFR30 [R/W] 00000000	PFR31 [R/W] 00000000	
000DA0 _H	PFR32 [R/W] 00000000	PFR33 [R/W] 00000000	PFR34 [R/W] 00000000	PFR35 [R/W] 00000000	
000DA4 _H	PFR36 [R/W] 000 - 00 - -	reserved	PFR38 [R/W] - - - - - 00	PFR39 [R/W] 00000000	
000DA8 _H	PFR40 [R/W] 00000000	reserved	reserved	reserved	
000DAC _H - 000DBC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
000DC0 _H	reserved	reserved	reserved	reserved	Extended Port Function register
000DC4 _H	reserved	reserved	reserved	reserved	
000DC8 _H	reserved	reserved	EPFR10 [R/W] -- 00 --- 0	reserved	
000DCC _H	EPFR12 [R/W] - 0 - - 0 - -	EPFR13 [R/W] - 0 - - 0 - -	EPFR14 [R/W] 00000000	EPFR15 [R/W] 00000000	
000DD0 _H	EPFR16 [R/W] 0000 - - - -	EPFR17 [R/W] 0000 - 000	EPFR18 [R/W] - 000 - 000	EPFR19 [R/W] - 0 - - - 0 - -	
000DD4 _H	EPFR20 [R/W] - 000 - 000	EPFR21 [R/W] - 0 - - - 0 - -	EPFR22 [R/W] - - - - 0 - 0 -	reserved	
000DD8 _H	reserved	reserved	EPFR26 [R/W] 00000000	EPFR27 [R/W] 00000000	
000DDC _H	EPFR28 [R/W] 0000 0 - 00	reserved	EPFR30 [R/W] 00000000	EPFR31 [R/W] - 000 - 000	
000DE0 _H	EPFR32 [R/W] 00000000	EPFR33 [R/W] 00000000	EPFR34 [R/W] 00000000	EPFR35 [R/W] 00000000	
000DE4 _H	reserved	reserved	reserved	reserved	
000DE8 _H	reserved	reserved	reserved	reserved	
000DEC _H - 000DFC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000E00 _H	PODR00 [R/W] 00000000	PODR01 [R/W] 00000000	PODR02 [R/W] 00000000	PODR03 [R/W] 00000000	Port Output Drive Strength control
000E04 _H	PODR04 [R/W] 00000000	PODR05 [R/W] 00000000	PODR06 [R/W] 00000000	PODR07 [R/W] 00000000	
000E08 _H	PODR08 [R/W] 00000000	PODR09 [R/W] 00000000	PODR10 [R/W] 00000000	PODR11 [R/W] 00000000	
000E0C _H	PODR12 [R/W] 00000000	PODR13 [R/W] 00000000	PODR14 [R/W] 00000000	PODR15 [R/W] 00000000	
000E10 _H	PODR16 [R/W] 00000000	PODR17 [R/W] 00000000	PODR18 [R/W] 00000000	PODR19 [R/W] 00000000	
000E14 _H	PODR20 [R/W] 00000000	PODR21 [R/W] 00000000	PODR22 [R/W] 00000000	PODR23 [R/W] 00000000	
000E18 _H	PODR24 [R/W] 00000000	PODR25 [R/W] 00000000	PODR26 [R/W] 00000000	PODR27 [R/W] 00000000	
000E1C _H	PODR28 [R/W] 00000000	PODR29 [R/W] 00000000	PODR30 [R/W] 00000000	PODR00 [R/W] 00000000	
000E20 _H	PODR32 [R/W] 00000000	PODR33 [R/W] 00000000	PODR34 [R/W] 00000000	PODR35 [R/W] 00000000	
000E24 _H	reserved	PODR37 [R/W] 00000000	reserved	reserved	
000E28 _H	reserved	reserved	reserved	reserved	
000E2C _H - 000E3C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
000E40 _H	PILR00 [R/W] 00000000	PILR01 [R/W] 00000000	PILR02 [R/W] 00000000	PILR03 [R/W] 00000000	Port Input Level selection register
000E44 _H	PILR04 [R/W] 00000000	PILR05 [R/W] 00000000	PILR06 [R/W] 00000000	PILR07 [R/W] 00000000	
000E48 _H	PILR08 [R/W] 00000000	PILR09 [R/W] 00000000	PILR10 [R/W] 00000000	PILR11 [R/W] 00000000	
000E4C _H	PILR12 [R/W] 00000000	PILR13 [R/W] 00000000	PILR14 [R/W] 00000000	PILR15 [R/W] 00000000	
000E50 _H	PILR16 [R/W] 00000000	PILR17 [R/W] 00000000	PILR18 [R/W] 00000000	PILR19 [R/W] 00000000	
000E54 _H	PILR20 [R/W] 00000000	PILR21 [R/W] 00000000	PILR22 [R/W] 00000000	PILR23 [R/W] 00000000	
000E58 _H	PILR24 [R/W] 00000000	PILR25 [R/W] 00000000	PILR26 [R/W] 00000000	PILR27 [R/W] 00000000	
000E5C _H	PILR28 [R/W] 00000000	PILR29 [R/W] 00000000	PILR30 [R/W] 00000000	PILR31 [R/W] 00000000	
000E60 _H	PILR32 [R/W] 00000000	PILR33 [R/W] 00000000	PILR34 [R/W] 00000000	PILR35 [R/W] 00000000	
000E64 _H	PILR36 [R/W] 00000000	PILR37 [R/W] 00000000	PILR38 [R/W] 00000000	PILR39 [R/W] 00000000	
000E68 _H	PILR40 [R/W] 00000000	reserved	reserved	reserved	
000E6C _H - 000E7C _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000E80 _H	EPILR00 [R/W] 00000000	EPILR01 [R/W] 00000000	EPILR02 [R/W] 00000000	EPILR03 [R/W] 00000000	Extended Port Input Level selection register
000E84 _H	EPILR04 [R/W] 00000000	EPILR05 [R/W] 00000000	EPILR06 [R/W] 00000000	EPILR07 [R/W] 00000000	
000E88 _H	EPILR08 [R/W] 00000000	EPILR09 [R/W] 00000000	EPILR10 [R/W] 00000000	EPILR11 [R/W] 00000000	
000E8C _H	EPILR12 [R/W] 00000000	EPILR13 [R/W] 00000000	EPILR14 [R/W] 00000000	EPILR15 [R/W] 00000000	
000E90 _H	EPILR16 [R/W] 00000000	EPILR17 [R/W] 00000000	EPILR18 [R/W] 00000000	EPILR19 [R/W] 00000000	
000E94 _H	EPILR20 [R/W] 00000000	EPILR21 [R/W] 00000000	EPILR22 [R/W] 00000000	EPILR23 [R/W] 00000000	
000E98 _H	EPILR24 [R/W] 00000000	EPILR25 [R/W] 00000000	EPILR26 [R/W] 00000000	EPILR27 [R/W] 00000000	
000E9C _H	EPILR28 [R/W] 00000000	EPILR29 [R/W] 00000000	EPILR30 [R/W] 00000000	EPILR31 [R/W] 00000000	
000EA0 _H	EPILR32 [R/W] 00000000	EPILR33 [R/W] 00000000	EPILR34 [R/W] 00000000	EPILR35 [R/W] 00000000	
000EA4 _H	EPILR36 [R/W] 000 - - - - -	EPILR37 [R/W] 00000000	reserved	reserved	
000EA8 _H	EPILR40 [R/W] 00000000	reserved	reserved	reserved	
000EAC _H - 000EBC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
000EC0 _H	PPER00 [R/W] 00000000	PPER01 [R/W] 00000000	PPER02 [R/W] 00000000	PPER03 [R/W] 00000000	Port Pull-Up/ Down Enable register
000EC4 _H	PPER04 [R/W] 00000000	PPER05 [R/W] 00000000	PPER06 [R/W] 00000000	PPER07 [R/W] 00000000	
000EC8 _H	PPER08 [R/W] 00000000	PPER09 [R/W] 00000000	PPER10 [R/W] 00000000	PPER11 [R/W] 00000000	
000ECC _H	PPER12 [R/W] 00000000	PPER13 [R/W] 00000000	PPER14 [R/W] 00000000	PPER15 [R/W] 00000000	
000ED0 _H	PPER16 [R/W] 00000000	PPER17 [R/W] 00000000	PPER18 [R/W] 00000000	PPER19 [R/W] 00000000	
000ED4 _H	PPER20 [R/W] 00000000	PPER21 [R/W] 00000000	PPER22 [R/W] 00000000	PPER23 [R/W] 00000000	
000ED8 _H	PPER24 [R/W] 00000000	PPER25 [R/W] 00000000	PPER26 [R/W] 00000000	PPER27 [R/W] 00000000	
000EDC _H	PPER28 [R/W] 00000000	PPER29 [R/W] 00000000	PPER30 [R/W] 00000000	PPER31 [R/W] 00000000	
000EE0 _H	PPER32 [R/W] 00000000	PPER33 [R/W] 00000000	PPER34 [R/W] 00000000	PPER35 [R/W] 00000000	
000EE4 _H	PPER36 [R/W] 00000000	PPER37 [R/W] 00000000	PPER38 [R/W] 00000000	PPER39 [R/W] 00000000	
000EE8 _H	PPER40 [R/W] 00000000	reserved	reserved	reserved	
000EEC _H - 000EFC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
000F00 _H	PPCR00 [R/W] 11111111	PPCR01 [R/W] 11111111	PPCR02 [R/W] 11111111	PPCR03 [R/W] 11111111	Port Pull-Up/ Down Control register
000F04 _H	PPCR04 [R/W] 11111111	PPCR05 [R/W] 11111111	PPCR06 [R/W] 11111111	PPCR07 [R/W] 11111111	
000F08 _H	PPCR08 [R/W] 11111111	PPCR09 [R/W] 11111111	PPCR10 [R/W] 11111111	PPCR11 [R/W] 11111111	
000F0C _H	PPCR12 [R/W] 11111111	PPCR13 [R/W] 11111111	PPCR14 [R/W] 11111111	PPCR15 [R/W] 11111111	
000F10 _H	PPCR16 [R/W] 11111111	PPCR17 [R/W] 11111111	PPCR18 [R/W] 11111111	PPCR19 [R/W] 11111111	
000F14 _H	PPCR20 [R/W] 11111111	PPCR21 [R/W] 11111111	PPCR22 [R/W] 11111111	PPCR23 [R/W] 11111111	
000F18 _H	PPCR24 [R/W] 11111111	PPCR25 [R/W] 11111111	PPCR26 [R/W] 11111111	PPCR27 [R/W] 11111111	
000F1C _H	PPCR28 [R/W] 11111111	PPCR29 [R/W] 11111111	PPCR30 [R/W] 11111111	PPCR31 [R/W] 11111111	
000F20 _H	PPCR32 [R/W] 11111111	PPCR33 [R/W] 11111111	PPCR34 [R/W] 11111111	PPCR35 [R/W] 11111111	
000F24 _H	PPCR36 [R/W] 11111111	PPCR37 [R/W] 11111111	PPCR38 [R/W] 11111111	PPCR39 [R/W] 11111111	
000F28 _H	PPCR40 [R/W] 11111111	reserved	reserved	reserved	
000F2C _H - 000F3C _H	reserved				
000F40 _H - 000FFC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
001000 _H	DMASA0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMAC
001004 _H	DMADA0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001008 _H	DMASA1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00100C _H	DMADA1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001010 _H	DMASA2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001014 _H	DMADA2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001018 _H	DMASA3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00101C _H	DMADA3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001020 _H	DMASA4 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001024 _H	DMADA4 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001028 _H - 01FFC _H	reserved				
002000 _H - 005FFC _H	MB91FV460B Instruction RAM/Flash Cache size is 16KB				Instruction RAM/ Flash Cache
006000 _H	DCCR [R/W] 00000000 0----- 00000000				MediaLB
006004 _H	SSCR [R/W] ----- 00000000				
006008 _H	SDCR [R/W] 00000000 00000000 00000000 00000000				
00600C _H	SMCR [R/W] ----- -1100000				
006010 _H to 006018 _H	Reserved				
00601C _H	VCCR [R] ----- 00000001 00000010 00000010				
006020 _H to 00602C _H	Reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
006030 _H	CICR [R/W] ----- -0000000 00000000				MediaLB
006034 _H to 00603C _H	Reserved				
006040 _H	CECR0 [R/W] 0000000- 00000000 00000000 00000000				
006044 _H	CSCR0 [R/W] 10----- ----0000 00000000 00000000				
006048 _H	CCBCR0 [R] 00000000 00000000 00000000 00000000				
00604C _H	CNBCR0 [R/W] 00000000 00000000 00000000 00000000				
006050 _H	CECR1 [R/W] 0000000- 00000000 00000000 00000000				
006054 _H	CSCR1 [R/W] 10----- ----0000 00000000 00000000				
006058 _H	CCBCR1 [R] 00000000 00000000 00000000 00000000				
00605C _H	CNBCR1 [R/W] 00000000 00000000 00000000 00000000				
006060 _H	CECR2 [R/W] 0000000- 00000000 00000000 00000000				
006064 _H	CSCR2 [R/W] 10----- ----0000 00000000 00000000				
006068 _H	CCBCR2 [R] 00000000 00000000 00000000 00000000				
00606C _H	CNBCR2 [R/W] 00000000 00000000 00000000 00000000				
006070 _H	CECR3 [R/W] 0000000- 00000000 00000000 00000000				
006074 _H	CSCR3 [R/W] 10----- ----0000 00000000 00000000				
006078 _H	CCBCR3 [R] 00000000 00000000 00000000 00000000				
00607C _H	CNBCR3 [R/W] 00000000 00000000 00000000 00000000				

Address	Register				Block
	+0	+1	+2	+3	
006080 _H	CECR4 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
006084 _H	CSCR4 [R/W] 10----- ----0000 00000000 00000000				
006088 _H	CCBCR4 [R] 00000000 00000000 00000000 00000000				
00608C _H	CNBCR4 [R/W] 00000000 00000000 00000000 00000000				
006090 _H	CECR5 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
006094 _H	CSCR5 [R/W] 10----- ----0000 00000000 00000000				
006098 _H	CCBCR5 [R] 00000000 00000000 00000000 00000000				
00609C _H	CNBCR5 [R/W] 00000000 00000000 00000000 00000000				
0060A0 _H	CECR6 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060A4 _H	CSCR6 [R/W] 10----- ----0000 00000000 00000000				
0060A8 _H	CCBCR6 [R] 00000000 00000000 00000000 00000000				
0060AC _H	CNBCR6 [R/W] 00000000 00000000 00000000 00000000				
0060B0 _H	CECR7 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060B4 _H	CSCR7 [R/W] 10----- ----0000 00000000 00000000				
0060B8 _H	CCBCR7 [R] 00000000 00000000 00000000 00000000				
0060BC _H	CNBCR7 [R/W] 00000000 00000000 00000000 00000000				
0060C0 _H	CECR8 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060C4 _H	CSCR8 [R/W] 10----- ----0000 00000000 00000000				
0060C8 _H	CCBCR8 [R] 00000000 00000000 00000000 00000000				
0060CC _H	CNBCR8 [R/W] 00000000 00000000 00000000 00000000				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0060D0 _H	CECR9 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060D4 _H	CSCR9 [R/W] 10----- ----0000 00000000 00000000				
0060D8 _H	CCBCR9 [R] 00000000 00000000 00000000 00000000				
0060DC _H	CNBCR9 [R/W] 00000000 00000000 00000000 00000000				
0060E0 _H	CECR10 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060E4 _H	CSCR10 [R/W] 10----- ----0000 00000000 00000000				
0060E8 _H	CCBCR10 [R] 00000000 00000000 00000000 00000000				
0060EC _H	CNBCR10 [R/W] 00000000 00000000 00000000 00000000				
0060F0 _H	CECR11 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
0060F4 _H	CSCR11 [R/W] 10----- ----0000 00000000 00000000				
0060F8 _H	CCBCR11 [R] 00000000 00000000 00000000 00000000				
0060FC _H	CNBCR11 [R/W] 00000000 00000000 00000000 00000000				
006100 _H	CECR12 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
006104 _H	CSCR12 [R/W] 10----- ----0000 00000000 00000000				
006108 _H	CCBCR12 [R] 00000000 00000000 00000000 00000000				
00610C _H	CNBCR12 [R/W] 00000000 00000000 00000000 00000000				
006110 _H	CECR13 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
006114 _H	CSCR13 [R/W] 10----- ----0000 00000000 00000000				
006118 _H	CCBCR13 [R] 00000000 00000000 00000000 00000000				
00611C _H	CNBCR13 [R/W] 00000000 00000000 00000000 00000000				

Address	Register				Block
	+0	+1	+2	+3	
006120 _H	CECR14 [R/W] 00000000- 00000000 00000000 00000000				MediaLB
006124 _H	CSCR14 [R/W] 10----- ----0000 00000000 00000000				
006128 _H	CCBCR14 [R] 00000000 00000000 00000000 00000000				
00612C _H	CNBCR14 [R/W] 00000000 00000000 00000000 00000000				
006130 _H to 00627F _H	Reserved				Reserved
006280 _H	LCBCR0 [R/W] 00000000 01000000 00000000 00000000				MediaLB
006284 _H	LCBCR1 [R/W] 00000000 01000000 00000000 00000001				
006288 _H	LCBCR2 [R/W] 00000000 01000000 00000000 00000010				
00628C _H	LCBCR3 [R/W] 00000000 01000000 00000000 00000011				
006290 _H	LCBCR4 [R/W] 00000000 01000000 00000000 00000100				
006294 _H	LCBCR5 [R/W] 00000000 01000000 00000000 00000101				
006298 _H	LCBCR6 [R/W] 00000000 01000000 00000000 00000110				
00629C _H	LCBCR7 [R/W] 00000000 01000000 00000000 00000111				
0062A0 _H	LCBCR8 [R/W] 00000000 01000000 00000000 00001000				
0062A4 _H	LCBCR9 [R/W] 00000000 01000000 00000000 00001001				
0062A8 _H	LCBCR10 [R/W] 00000000 01000000 00000000 00001010				
0062AC _H	LCBCR11 [R/W] 00000000 01000000 00000000 00001011				
0062B0 _H	LCBCR12 [R/W] 00000000 01000000 00000000 00001100				
0062B4 _H	LCBCR13 [R/W] 00000000 01000000 00000000 00001101				
0062B8 _H	LCBCR14 [R/W] 00000000 01000000 00000000 00001110				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0062BC _H to 00630F _H	Reserved				Reserved
006310 _H	I2SSCR [R/W] 00100000 0----000		I2SRSR [R/W] ----- 00000000		I2S 0 to 9
006314 _H	Reserved				
006318 _H	I2SSCR0 [R/W] ----0000 0--00000		I2SBT0 [R] ----0000	I2SBCR0 [R/W] ---00000	I2S 0
00631C _H	LTDT0 [R/W] 00000000 00000000		RTDT0 [R/W] 00000000 00000000		
006320 _H	I2SSCR1 [R/W] ----0000 0--00000		I2SBT1 [R] ----0000	I2SBCR1 [R/W] ---00000	I2S 1
006324 _H	LTDT1 [R/W] 00000000 00000000		RTDT1 [R/W] 00000000 00000000		
006328 _H	I2SSCR2 [R/W] ----0000 0--00000		I2SBT2 [R] ----0000	I2SBCR2 [R/W] ---00000	I2S 2
00632C _H	LTDT2 [R/W] 00000000 00000000		RTDT2 [R/W] 00000000 00000000		
006330 _H	I2SSCR3 [R/W] ----0000 0--00000		I2SBT3 [R] ----0000	I2SBCR3 [R/W] ---00000	I2S 3
006334 _H	LTDT3 [R/W] 00000000 00000000		RTDT3 [R/W] 00000000 00000000		
006338 _H	I2SSCR4 [R/W] ----0000 0--00000		I2SBT4 [R] ----0000	I2SBCR4 [R/W] ---00000	I2S 4
00633C _H	LTDT4 [R/W] 00000000 00000000		RTDT4 [R/W] 00000000 00000000		
006340 _H	I2SSCR5 [R/W] ----0000 0--00000		I2SBT5 [R] ----0000	I2SBCR5 [R/W] ---00000	I2S 5
006344 _H	LTDT5 [R/W] 00000000 00000000		RTDT5 [R/W] 00000000 00000000		
006348 _H	I2SSCR6 [R/W] ----0000 0--00000		I2SBT6 [R] ----0000	I2SBCR6 [R/W] ---00000	I2S 6
00634C _H	LTDT6 [R/W] 00000000 00000000		RTDT6 [R/W] 00000000 00000000		
006350 _H	I2SSCR7 [R/W] ----0000 0--00000		I2SBT7 [R] ----0000	I2SBCR7 [R/W] ---00000	I2S 7
006354 _H	LTDT7 [R/W] 00000000 00000000		RTDT7 [R/W] 00000000 00000000		
006358 _H	I2SSCR8 [R/W] ----0000 0--00000		I2SBT8 [R] ----0000	I2SBCR8 [R/W] ---00000	I2S 8
00635C _H	LTDT8 [R/W] 00000000 00000000		RTDT8 [R/W] 00000000 00000000		

Address	Register				Block
	+0	+1	+2	+3	
006360 _H	I2SSCR9 [R/W] ----0000 0--00000		I2SBT9 [R] ----0000	I2SBCR9 [R/W] ---00000	I2S 9
006364 _H	LTDT9 [R/W] 00000000 00000000		RTDT9 [R/W] 00000000 00000000		
006368 _H to 00640C _H	Reserved				Reserved
006410 _H	BUFAR0 [R/W] 00000000 00000000 00000000 00000000				MediaLB
006414 _H	BUFAR1 [R/W] 00000000 00000000 00000000 00000000				
006418 _H	BUFAR2 [R/W] 00000000 00000000 00000000 00000000				
00641C _H	BUFAR3 [R/W] 00000000 00000000 00000000 00000000				
006420 _H	BUFAR4 [R/W] 00000000 00000000 00000000 00000000				
006424 _H	BUFAR5 [R/W] 00000000 00000000 00000000 00000000				
006428 _H	BUFAR6 [R/W] 00000000 00000000 00000000 00000000				
00642C _H	BUFAR7 [R/W] 00000000 00000000 00000000 00000000				
006430 _H	MSTD [R/W] -0000000 00000000		MBSYNC [R] XXXXXXXX XXXXXXXX		MediaLB
006434 _H	BUFAR8 [R/W] 00000000 00000000 00000000 00000000				MediaLB
006438 _H	BUFAR9 [R/W] 00000000 00000000 00000000 00000000				
00643C _H	BUFAR10 [R/W] 00000000 00000000 00000000 00000000				
006440 _H	BUFAR11 [R/W] 00000000 00000000 00000000 00000000				
006444 _H	BUFAR12 [R/W] 00000000 00000000 00000000 00000000				
006448 _H	BUFAR13 [R/W] 00000000 00000000 00000000 00000000				
00644C _H	BUFAR14 [R/W] 00000000 00000000 00000000 00000000				
006450 _H to 00649C _H	Reserved				Reserved

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
0064A0 _H	ASLR [R/W] 00000000 00000000 00000000 00000000				FIFO buffer
0064A4 _H	BUFDCR [R/W] 00000000 00000000 00000000 00000000				
0064A8 _H	BUFIER [R/W] 00000000 00000000 -00000000 00000000				
0064AC _H	BUFSR [R/W] 00000000 00000000 -00000000 00000000				
0064B0 _H	BUFER [R/W] 00000000 00000000 -00000000 00000000				
0064B4 _H	BUFRST [R/W] 00000000 00000000 -00000000 00000000				
0064B8 _H , 0064BC _H	Reserved				
0064C0 _H	BUFCT0 [R/W] ----0000 00000000 ----0000 00000000				FIFO buffer
0064C4 _H	BUFCT1 [R/W] ----0000 00000000 ----0000 00000000				
0064C8 _H	BUFCT2 [R/W] ----0000 00000000 ----0000 00000000				
0064CC _H	BUFCT3 [R/W] ----0000 00000000 ----0000 00000000				
0064D0 _H	BUFCT4 [R/W] ----0000 00000000 ----0000 00000000				
0064D4 _H	BUFCT5 [R/W] ----0000 00000000 ----0000 00000000				
0064D8 _H	BUFCT6 [R/W] ----0000 00000000 ----0000 00000000				
0064DC _H	BUFCT7 [R/W] ----0000 00000000 ----0000 00000000				
0064E0 _H to 0064FC _H	Reserved				

Address	Register				Block
	+0	+1	+2	+3	
006500 _H	BUF0CR [R/W] 00000000 00000000 000----0 00000000				FIFO buffer
006504 _H	BUF1CR [R/W] 00000000 00000000 000----0 00000000				
006508 _H	BUF2CR [R/W] 00000000 00000000 000----0 00000000				
00650C _H	BUF3CR [R/W] 00000000 00000000 000----0 00000000				
006510 _H	BUF4CR [R/W] 00000000 00000000 000----0 00000000				
006514 _H	BUF5CR [R/W] 00000000 00000000 000----0 00000000				
006518 _H	BUF6CR [R/W] 00000000 00000000 000----0 00000000				
00651C _H	BUF7CR [R/W] 00000000 00000000 000----0 00000000				
006520 _H	BUF8CR [R/W] 00000000 00000000 000----0 00000000				
006524 _H	BUF9CR [R/W] 00000000 00000000 000----0 00000000				
006528 _H	BUF10CR [R/W] 00000000 00000000 000----0 00000000				
00652C _H	BUF11CR [R/W] 00000000 00000000 000----0 00000000				
006530 _H	BUF12CR [R/W] 00000000 00000000 000----0 00000000				
006534 _H	BUF13CR [R/W] 00000000 00000000 000----0 00000000				
006538 _H	BUF14CR [R/W] 00000000 00000000 000----0 00000000				
00653C _H to 00657C _H	Reserved				Reserved

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
006580 _H	BUF0DTR [R/W] 00000000 00000000 00000000 00000000				FIFO buffer
006584 _H	BUF1DTR [R/W] 00000000 00000000 00000000 00000000				
006588 _H	BUF2DTR [R/W] 00000000 00000000 00000000 00000000				
00658C _H	BUF3DTR [R/W] 00000000 00000000 00000000 00000000				
006590 _H	BUF4DTR [R/W] 00000000 00000000 00000000 00000000				
006594 _H	BUF5DTR [R/W] 00000000 00000000 00000000 00000000				
006598 _H	BUF6DTR [R/W] 00000000 00000000 00000000 00000000				
00659C _H	BUF7DTR [R/W] 00000000 00000000 00000000 00000000				
0065A0 _H	BUF8DTR [R/W] 00000000 00000000 00000000 00000000				
0065A4 _H	BUF9DTR [R/W] 00000000 00000000 00000000 00000000				
0065A8 _H	BUF10DTR [R/W] 00000000 00000000 00000000 00000000				
0065AC _H	BUF11DTR [R/W] 00000000 00000000 00000000 00000000				
0065B0 _H	BUF12DTR [R/W] 00000000 00000000 00000000 00000000				
0065B4 _H	BUF13DTR [R/W] 00000000 00000000 00000000 00000000				
0065B8 _H	BUF14DTR [R/W] 00000000 00000000 00000000 00000000				
0065BC _H	Reserved				Reserved
006600 _H	MLBINTR [R] -0000-00 00000000 -----00 00000000				MediaLB I2S FIFO buffer Interrupt
006604 _H	BUFINTCH [R] 0---0000		Reserved		
006608 _H	BUFPRI0001 [R/ W] 11101101	BUFPRI0203 [R/ W] 11001011	BUFPRI0405 [R/ W] 10101001	BUFPRI0607 [R/ W] 10000111	
00660C _H	BUFPRI0809 [R/ W] 01100101	BUFPRI1011 [R/ W] 01000011	BUFPRI1213 [R/ W] 00100001	BUFPRI14 [R/W] 0000----	

Address	Register				Block
	+0	+1	+2	+3	
006610 _H to 006FFC _H	Reserved				Reserved
007000 _H	FMCS [R/W] 01101000	FMCR [R] ---- 0000	FCHCR [R/W] ----- 00 10000011		Flash Memory/ Flash Cache Control Register
007004 _H	FMWT [R/W] 11111111 01011101 ¹⁰		FMWT2 [R/W,R] - 101 ----	FMPs [R/W] ----- 000	
007008 _H	FMAC [R] 00000000 00000000 00000000 00000000				
00700C _H	FCHA0 [R/W] ----- --- 00000 00000000 00000000				F-Cache Non- cacheable area setting Register
007010 _H	FCHA1 [R/W] ----- --- 00000 00000000 00000000				
007014 _H - 00701C _H	reserved				
007020 _H	BFMCS [R/W] 0 - 10 1000	BFMCR [R] ---- 0000	reserved		Boot Flash Control Register (MB91FV460B)
007024 _H	BFMWT [R/W] 00100101 00010011		BFMWT2 [R/ W,R] - 001 ----	reserved	
007028 _H - 0070FC _H	reserved				
007100 _H	FSCR0 [R/W, R] 11111111 11111111 11111111 11111111				Flash Security CRC Control register
007104 _H	FSCR1 [R , R/W] 0 - - - 0001 00000000 00000000 00000000				
007108 _H - 007110 _H	reserved				
007114 _H	DFCS [R/W] 0000 000X	DFWC [R/W] --- 0 0000	DFWS [R/W,R] 0000 0000	reserved	Data Flash Con- trol register
007118 _H	DFSCR0 [R/W, R] 11111111 11111111 11111111 11111111				Data Flash Security CRC control
00711C _H	DFSCR1 [R , R/W] 0 - - - 0000 00000000 00000000 00000000				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
007120 _H	CRCPOLY [R/W] 00000100 11000001 00011101 10110111				Programmable CRC Module
007124 _H	CRCSEED [R/W] 11111111 11111111 11111111 11111111				
007128 _H	CRCWR [R/W] 00000000 00000000 00000000 00000000				
00712C _H	CRCRD [R/W] 00000000 00000000 00000000 00000000				
007130 _H	CRCFXOR [R/W] 00000000 00000000 00000000 00000000				
007134 _H	CRCCFG0 [R/W] 00000000	CRCCFG1 [R/W] 11100000	CRCCFG2 [R/W] 00000001	reserved	
007138 _H - 0071FC _H	reserved				
007200 _H	RHCTRL[R/W] 00 -- 0000 - - - - - 0000 - 000 - - - - -				APIX® * 11 Remotehandler Control
007204 _H	reserved				
007208 _H	CHCTRL0[R/W] - - - - 0 - - - 000 - - - 00 00000000 - - 00 - - 111				APIX® Contol/Status
00720C _H	CHSTAT0[R] 00000000 - - - - - 00000000 00000000				
007210 _H	CHWDG0[R/W] 00 - - 0000 00000000 xxxxxxxx xxxxxxxx				
007214 _H	CHCTRL1[R/W] - - - - 0 - - - 000 - - - 00 00000000 - - 00 - - 111				
007218 _H	CHSTAT1[R] 00000000 - - - - - 00000000 00000000				
00721C _H	CHWDG1[R/W] 00 - - 0000 00000000 xxxxxxxx xxxxxxxx				

Address	Register				Block
	+0	+1	+2	+3	
007220 _H	TBCTRL00[R/W] - - - 00000 0000 - 000		TBCTRL01[R/W] - - - 00000 0000 - 000		APIX® Remotehandler Transactionbuf- fer Control
007224 _H	TBCTRL02[R/W] - - - 00000 0000 - 000		TBCTRL03[R/W] - - - 00000 0000 - 000		
007228 _H	TBCTRL04[R/W] - - - 00000 0000 - 000		TBCTRL05[R/W] - - - 00000 0000 - 000		
00722C _H	TBCTRL06[R/W] - - - 00000 0000 - 000		TBCTRL07[R/W] - - - 00000 0000 - 000		
007230 _H	TBCTRL08[R/W] - - - 00000 0000 - 000		TBCTRL09[R/W] - - - 00000 0000 - 000		
007234 _H	TBCTRL10[R/W] - - - 00000 0000 - 000		TBCTRL11[R/W] - - - 00000 0000 - 000		
007238 _H	TBCTRL12[R/W] - - - 00000 0000 - 000		TBCTRL13[R/W] - - - 00000 0000 - 000		
00723C _H	TBCTRL14[R/W] - - - 00000 0000 - 000		TBCTRL15[R/W] - - - 00000 0000 - 000		
007240 _H	TBIRQ 00000000 00000000		reserved		APIX® Remote- handler Interrupt
007244 _H - 00724C _H	reserved				
007250 _H	TFCRTL00[R/W] 00 - 00000	TFIDX00[R/W] 00000000	TFCRTL01[R/W] 00 - 00000	TFIDX01[R/W] 00000000	APIX® Transactionframe
007254 _H	TFCRTL02[R/W] 00 - 00000	TFIDX02[R/W] 00000000	TFCRTL03[R/W] 00 - 00000	TFIDX03[R/W] 00000000	
007258 _H	TFCRTL04[R/W] 00 - 00000	TFIDX04[R/W] 00000000	TFCRTL05[R/W] 00 - 00000	TFIDX05[R/W] 00000000	
00725C _H	TFCRTL06[R/W] 00 - 00000	TFIDX06[R/W] 00000000	TFCRTL07[R/W] 00 - 00000	TFIDX07[R/W] 00000000	
007260 _H	TFCRTL08[R/W] 00 - 00000	TFIDX08[R/W] 00000000	TFCRTL09[R/W] 00 - 00000	TFIDX09[R/W] 00000000	
007264 _H	TFCRTL10[R/W] 00 - 00000	TFIDX10[R/W] 00000000	TFCRTL11[R/W] 00 - 00000	TFIDX11[R/W] 00000000	
007268 _H	TFCRTL12[R/W] 00 - 00000	TFIDX12[R/W] 00000000	TFCRTL13[R/W] 00 - 00000	TFIDX13[R/W] 00000000	
00726C _H	TFCRTL14[R/W] 00 - 00000	TFIDX14[R/W] 00000000	TFCRTL15[R/W] 00 - 00000	TFIDX15[R/W] 00000000	

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
007270 _H	TFADDR00[R/W] ----- 0000 00000000 00000000				APIX® Transactionframe
007274 _H	TFADDR01[R/W] ----- 0000 00000000 00000000				
007278 _H	TFADDR02[R/W] ----- 0000 00000000 00000000				
00727C _H	TFADDR03[R/W] ----- 0000 00000000 00000000				
007280 _H	TFADDR04[R/W] ----- 0000 00000000 00000000				
007284 _H	TFADDR05[R/W] ----- 0000 00000000 00000000				
007288 _H	TFADDR06[R/W] ----- 0000 00000000 00000000				
00728C _H	TFADDR07[R/W] ----- 0000 00000000 00000000				
007290 _H	TFADDR08[R/W] ----- 0000 00000000 00000000				
007294 _H	TFADDR09[R/W] ----- 0000 00000000 00000000				
007298 _H	TFADDR10[R/W] ----- 0000 00000000 00000000				
00729C _H	TFADDR11[R/W] ----- 0000 00000000 00000000				
0072A0 _H	TFADDR12[R/W] ----- 0000 00000000 00000000				APIX® Transactionframe
0072A4 _H	TFADDR13[R/W] ----- 0000 00000000 00000000				
0072A8 _H	TFADDR14[R/W] ----- 0000 00000000 00000000				
0072AC _H	TFADDR15[R/W] ----- 0000 00000000 00000000				

Address	Register				Block
	+0	+1	+2	+3	
0072B0 _H	TFDATA00[R/W] 00000000 00000000 00000000 00000000				APIX® Transactionframe
0072B4 _H	TFDATA01[R/W] 00000000 00000000 00000000 00000000				
0072B8 _H	TFDATA02[R/W] 00000000 00000000 00000000 00000000				
0072BC _H	TFDATA03[R/W] 00000000 00000000 00000000 00000000				
0072C0 _H	TFDATA04[R/W] 00000000 00000000 00000000 00000000				
0072C4 _H	TFDATA05[R/W] 00000000 00000000 00000000 00000000				
0072C8 _H	TFDATA06[R/W] 00000000 00000000 00000000 00000000				
0072CC _H	TFDATA07[R/W] 00000000 00000000 00000000 00000000				
0072D0 _H	TFDATA08[R/W] 00000000 00000000 00000000 00000000				
0072D4 _H	TFDATA09[R/W] 00000000 00000000 00000000 00000000				
0072D8 _H	TFDATA10[R/W] 00000000 00000000 00000000 00000000				
0072DC _H	TFDATA11[R/W] 00000000 00000000 00000000 00000000				
0072E0 _H	TFDATA12[R/W] 00000000 00000000 00000000 00000000				
0072E4 _H	TFDATA13[R/W] 00000000 00000000 00000000 00000000				
0072E8 _H	TFDATA14[R/W] 00000000 00000000 00000000 00000000				
0072EC _H	TFDATA15[R/W] 00000000 00000000 00000000 00000000				
0072F0 _H	EVCTRL ----- 0 0 - 000000 00000000 00000000				APIX® Remotehandler Eventcontrol
0072F4 _H	reserved				
0072F8 _H	EVBUF0[R/W] ----- 0 00000000 -----				APIX® Remotehandler Eventqueue
0072FC _H	EVBUF1[R/W] 00000000 00000000 00000000 00000000				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
007300 _H	APCFG00[R/W] 00000000 00110000 00000000 10010000				APIX® Configuration
007304 _H	APCFG01[R/W] 11110000 00000000 00000000 01001000				
007308 _H	APCFG02[R/W] 00000010 00000010 01000000 - - - - -				
00730C _H	APCFG03[R/W] 00100110 10100000 10011010 00 - - - 000				
007310 _H	APCFG10[R/W] 00000000 00110000 00000000 10010000				
007314 _H	APCFG11[R/W] 11110000 00000000 00000000 01001000				
007318 _H	APCFG12[R/W] 00000010 00000010 01000000 - - - - -				
00731C _H	APCFG13[R/W] 00100110 10100100 10011010 00 - - - 000				
007320 _H	MODULEID [R] 0 - - - - - ***** 12				APIX® Version
007324 _H - 0073FC _H	reserved				
007400 _H	HCNT [R/W] 0000 0001 0000 0000		HERR [R/W] 0000 0011	HIRQ [R/W] 0000 0000	USB 0
007404 _H	HFCOMP [R/W] 0000 0000	HSTATE [R/W] - 001 0010	HRTIMER1 [R/W] 0000 0000	HRTIMER0 [R/W] 0000 0000	
007408 _H	HADR [R/W] - 000 0000	HRTIMER2 [R/W] - - - - - 00	HEOF [R/W] - - 00 0000 0000 0000		
00740C _H	HFRAME [R/W] - - - - - 000 0000 0000		reserved	HTOKEN [R/W] 0000 0000	
007410 _H	TEST [R/W] 000 - 0000	UDCC [R/W] 1010 0000	EP0C [R/W] - - - - 000- -100 0000		
007414 _H	EP1C [R/W] 0110 0001 0000 0000		EP2C [R/W] 0110 0000 0100 0000		
007418 _H	EP3C [R/W] 0110 0000 0100 0000		EP4C [R/W] 0110 0000 0100 0000		
00741C _H	EP5C [R/W] 0110 0000 0100 0000		TMSP [R/W] 0000 0000 0000 0000		

Address	Register				Block
	+0	+1	+2	+3	
007420 _H	UDCIE [R/W] 0000 0000	UDCS [R/W] 0000 0000	EP0IS [R/W] 100 - - 10 - - 100 0000		USB 0
007424 _H	EP0OS [R/W] 100 - - 00 - - 100 0000		EP1S [R/W] 1000 0001 0000 0000		
007428 _H	EP2S [R/W] 1000 000- - 100 0000		EP3S [R/W] 1000 000- - 100 0000		
00742C _H	EP4S [R/W] 1000 000- - 100 0000		EP5S [R/W] 1000 000- - 100 0000		
007430 _H	EP0DT [R/W] xxxx xxxx xxxx xxxx		EP1DT [R/W] xxxx xxxx xxxx xxxx		
007434 _H	EP2DT [R/W] xxxx xxxx xxxx xxxx		EP3DT [R/W] xxxx xxxx xxxx xxxx		
007438 _H	EP4DT [R/W] xxxx xxxx xxxx xxxx		EP5DT [R/W] xxxx xxxx xxxx xxxx		
00743C _H - 0077FC _H	reserved				
007800 _H - 007FFC _H	MB91FV460B: Fallback Boot ROM for serial programming mode 2 KByte				Mini Boot ROM
008000 _H - 00BFFC _H	MB91FV460B: Boot Flash 16 KByte 008000H to 00BFFCH, other devices: Boot-ROM 4 KByte 00B000H to 00BFFCH (instruction access is 1 wait cycle, data access is 1 wait cycle)				Boot ROM area
00C000 _H	CTRLR0 [R/W] 00000000 00000001		STATR0 [R/W] 00000000 00000000		CAN 0 Control register
00C004 _H	ERRCNT0 [R] 00000000 00000000		BTR0 [R/W] 00100011 00000001		
00C008 _H	INTR0 [R] 00000000 00000000		TESTR0 [R/W] 00000000 X0000000		
00C00C _H	BRPE0 [R/W] 00000000 00000000		reserved		
00C010 _H	IF1CREQ0 [R/W] 00000000 00000001		IF1CMSK0 [R/W] 00000000 00000000		CAN 0 IF 1 Register
00C014 _H	IF1MSK20 [R/W] 11111111 11111111		IF1MSK10 [R/W] 11111111 11111111		
00C018 _H	IF1ARB20 [R/W] 00000000 00000000		IF1ARB10 [R/W] 00000000 00000000		
00C01C _H	IF1MCTR0 [R/W] 00000000 00000000		reserved		
00C020 _H	IF1DTA10 [R/W] 00000000 00000000		IF1DTA20 [R/W] 00000000 00000000		
00C024 _H	IF1DTB10 [R/W] 00000000 00000000		IF1DTB20 [R/W] 00000000 00000000		

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C028 _H	reserved				
00C02C _H	reserved				
00C030 _H	IF1DTA20 [R/W] 00000000 00000000		IF1DTA10 [R/W] 00000000 00000000		CAN 0 IF 1 Register mirror
00C034 _H	IF1DTB20 [R/W] 00000000 00000000		IF1DTB10 [R/W] 00000000 00000000		
00C038 _H	reserved				
00C03C _H	reserved				
00C040 _H	IF2CREQ0 [R/W] 00000000 00000001		IF2CMSK0 [R/W] 00000000 00000000		CAN 0 IF 2 Register
00C044 _H	IF2MSK20 [R/W] 11111111 11111111		IF2MSK10 [R/W] 11111111 11111111		
00C048 _H	IF2ARB20 [R/W] 00000000 00000000		IF2ARB10 [R/W] 00000000 00000000		
00C04C _H	IF2MCTR0 [R/W] 00000000 00000000		reserved		
00C050 _H	IF2DTA10 [R/W] 00000000 00000000		IF2DTA20 [R/W] 00000000 00000000		
00C054 _H	IF2DTB10 [R/W] 00000000 00000000		IF2DTB20 [R/W] 00000000 00000000		
00C058 _H	reserved				
00C05C _H	reserved				
00C060 _H	IF2DTA20 [R/W] 00000000 00000000		IF2DTA10 [R/W] 00000000 00000000		CAN 0 IF 2 Register mirror
00C064 _H	IF2DTB20 [R/W] 00000000 00000000		IF2DTB10 [R/W] 00000000 00000000		
00C068 _H - 00C07C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C080 _H	TREQR20 [R] 00000000 00000000		TREQR10 [R] 00000000 00000000		CAN 0 Status Flags
00C084 _H	TREQR40 [R] 00000000 00000000		TREQR30 [R] 00000000 00000000		
00C088 _H	TREQR60 [R] 00000000 00000000		TREQR50 [R] 00000000 00000000		
00C08C _H	TREQR80 [R] 00000000 00000000		TREQR70 [R] 00000000 00000000		
00C090 _H	NEWDT20 [R] 00000000 00000000		NEWDT10 [R] 00000000 00000000		
00C094 _H	NEWDT40 [R] 00000000 00000000		NEWDT30 [R] 00000000 00000000		
00C098 _H	NEWDT60 [R] 00000000 00000000		NEWDT50 [R] 00000000 00000000		
00C09C _H	NEWDT80 [R] 00000000 00000000		NEWDT70 [R] 00000000 00000000		
00C0A0 _H	INTPND20 [R] 00000000 00000000		INTPND10 [R] 00000000 00000000		
00C0A4 _H	INTPND40 [R] 00000000 00000000		INTPND30 [R] 00000000 00000000		
00C0A8 _H	INTPND60 [R] 00000000 00000000		INTPND50 [R] 00000000 00000000		
00C0AC _H	INTPND80 [R] 00000000 00000000		INTPND70 [R] 00000000 00000000		
00C0B0 _H	MSGVAL20 [R] 00000000 00000000		MSGVAL10 [R] 00000000 00000000		
00C0B4 _H	MSGVAL40 [R] 00000000 00000000		MSGVAL30 [R] 00000000 00000000		
00C0B8 _H	MSGVAL60 [R] 00000000 00000000		MSGVAL50 [R] 00000000 00000000		
00C0BC _H	MSGVAL80 [R] 00000000 00000000		MSGVAL70 [R] 00000000 00000000		
00C0C0 _H - 00C0FC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C100 _H	CTRLR1 [R/W] 00000000 00000001		STATR1 [R/W] 00000000 00000000		CAN 1 Control Register
00C104 _H	ERRCNT1 [R] 00000000 00000000		BTR1 [R/W] 00100011 00000001		
00C108 _H	INTR1 [R] 00000000 00000000		TESTR1 [R/W] 00000000 X0000000		
00C10C _H	BRPE1 [R/W] 00000000 00000000		reserved		
00C110 _H	IF1CREQ1 [R/W] 00000000 00000001		IF1CMSK1 [R/W] 00000000 00000000		CAN 1 IF 1 Register
00C114 _H	IF1MSK21 [R/W] 11111111 11111111		IF1MSK11 [R/W] 11111111 11111111		
00C118 _H	IF1ARB21 [R/W] 00000000 00000000		IF1ARB11 [R/W] 00000000 00000000		
00C11C _H	IF1MCTR1 [R/W] 00000000 00000000		reserved		
00C120 _H	IF1DTA11 [R/W] 00000000 00000000		IF1DTA21 [R/W] 00000000 00000000		
00C124 _H	IF1DTB11 [R/W] 00000000 00000000		IF1DTB21 [R/W] 00000000 00000000		
00C128 _H	reserved				
00C12C _H	reserved				
00C130 _H	IF1DTA21 [R/W] 00000000 00000000		IF1DTA11 [R/W] 00000000 00000000		CAN 1 IF 1 Register mirror
00C134 _H	IF1DTB21 [R/W] 00000000 00000000		IF1DTB11 [R/W] 00000000 00000000		
00C138 _H	reserved				
00C13C _H	reserved				
00C140 _H	IF2CREQ1 [R/W] 00000000 00000001		IF2CMSK1 [R/W] 00000000 00000000		CAN 1 IF 2 Register
00C144 _H	IF2MSK21 [R/W] 11111111 11111111		IF2MSK11 [R/W] 11111111 11111111		
00C148 _H	IF2ARB21 [R/W] 00000000 00000000		IF2ARB11 [R/W] 00000000 00000000		
00C14C _H	IF2MCTR1 [R/W] 00000000 00000000		reserved		
00C150 _H	IF2DTA11 [R/W] 00000000 00000000		IF2DTA21 [R/W] 00000000 00000000		
00C154 _H	IF2DTB11 [R/W] 00000000 00000000		IF2DTB21 [R/W] 00000000 00000000		
00C158 _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C15C _H	reserved				
00C160 _H	IF2DTA21 [R/W] 00000000 00000000		IF2DTA11 [R/W] 00000000 00000000		CAN 1 IF 2 Register mirror
00C164 _H	IF2DTB21 [R/W] 00000000 00000000		IF2DTB11 [R/W] 00000000 00000000		
00C168 _H - 00C17C _H	reserved				
00C180 _H	TREQR21 [R] 00000000 00000000		TREQR11 [R] 00000000 00000000		CAN 1 Status Flags
00C184 _H	TREQR41 [R] 00000000 00000000		TREQR31 [R] 00000000 00000000		
00C188 _H	TREQR61 [R] 00000000 00000000		TREQR51 [R] 00000000 00000000		
00C18C _H	TREQR81 [R] 00000000 00000000		TREQR71 [R] 00000000 00000000		
00C190 _H	NEWDT21 [R] 00000000 00000000		NEWDT11 [R] 00000000 00000000		
00C194 _H	NEWDT41 [R] 00000000 00000000		NEWDT31 [R] 00000000 00000000		
00C198 _H	NEWDT61 [R] 00000000 00000000		NEWDT51 [R] 00000000 00000000		
00C19C _H	NEWDT81 [R] 00000000 00000000		NEWDT71 [R] 00000000 00000000		
00C1A0 _H	INTPND21 [R] 00000000 00000000		INTPND11 [R] 00000000 00000000		
00C1A4 _H	INTPND41 [R] 00000000 00000000		INTPND31 [R] 00000000 00000000		
00C1A8 _H	INTPND61 [R] 00000000 00000000		INTPND51 [R] 00000000 00000000		
00C1AC _H	INTPND81 [R] 00000000 00000000		INTPND71 [R] 00000000 00000000		
00C1B0 _H	MSGVAL21 [R] 00000000 00000000		MSGVAL11 [R] 00000000 00000000		
00C1B4 _H	MSGVAL41 [R] 00000000 00000000		MSGVAL31 [R] 00000000 00000000		
00C1B8 _H	MSGVAL61 [R] 00000000 00000000		MSGVAL51 [R] 00000000 00000000		
00C1BC _H	MSGVAL81 [R] 00000000 00000000		MSGVAL71 [R] 00000000 00000000		

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C1C0 _H - 00C1FC _H	reserved				
00C200 _H	CTRLR2 [R/W] 00000000 00000001		STATR2 [R/W] 00000000 00000000		CAN 2 Control Register
00C204 _H	ERRCNT2 [R] 00000000 00000000		BTR2 [R/W] 00100011 00000001		
00C208 _H	INTR2 [R] 00000000 00000000		TESTR2 [R/W] 00000000 X0000000		
00C20C _H	BRPE2 [R/W] 00000000 00000000		CBSYNC2 ¹³		
00C210 _H	IF1CREQ2 [R/W] 00000000 00000001		IF1CMSK2 [R/W] 00000000 00000000		CAN 2 IF 1 Register
00C214 _H	IF1MSK22 [R/W] 11111111 11111111		IF1MSK12 [R/W] 11111111 11111111		
00C218 _H	IF1ARB22 [R/W] 00000000 00000000		IF1ARB12 [R/W] 00000000 00000000		
00C21C _H	IF1MCTR2 [R/W] 00000000 00000000		reserved		
00C220 _H	IF1DTA12 [R/W] 00000000 00000000		IF1DTA22 [R/W] 00000000 00000000		
00C224 _H	IF1DTB12 [R/W] 00000000 00000000		IF1DTB22 [R/W] 00000000 00000000		
00C228 _H	reserved				
00C22C _H	reserved				
00C230 _H	IF1DTA22 [R/W] 00000000 00000000		IF1DTA12 [R/W] 00000000 00000000		CAN 2 IF 1 Register mirror
00C234 _H	IF1DTB22 [R/W] 00000000 00000000		IF1DTB12 [R/W] 00000000 00000000		
00C238 _H	reserved				
00C23C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C240 _H	IF2CREQ2 [R/W] 00000000 00000001		IF2CMSK2 [R/W] 00000000 00000000		CAN 2 IF 2 Register
00C244 _H	IF2MSK22 [R/W] 11111111 11111111		IF2MSK12 [R/W] 11111111 11111111		
00C248 _H	IF2ARB22 [R/W] 00000000 00000000		IF2ARB12 [R/W] 00000000 00000000		
00C24C _H	IF2MCTR2 [R/W] 00000000 00000000		reserved		
00C250 _H	IF2DTA12 [R/W] 00000000 00000000		IF2DTA22 [R/W] 00000000 00000000		
00C254 _H	IF2DTB12 [R/W] 00000000 00000000		IF2DTB22 [R/W] 00000000 00000000		
00C258 _H	reserved				
00C25C _H	reserved				
00C260 _H	IF2DTA22 [R/W] 00000000 00000000		IF2DTA12 [R/W] 00000000 00000000		CAN 2 IF 2 Register mirror
00C264 _H	IF2DTB22 [R/W] 00000000 00000000		IF2DTB12 [R/W] 00000000 00000000		
00C268 _H - 00C27C _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C280 _H	TREQR22 [R] 00000000 00000000		TREQR12 [R] 00000000 00000000		CAN 2 Status Flags
00C284 _H	TREQR42 [R] 00000000 00000000		TREQR32 [R] 00000000 00000000		
00C288 _H	TREQR62 [R] 00000000 00000000		TREQR52 [R] 00000000 00000000		
00C28C _H	TREQR82 [R] 00000000 00000000		TREQR72 [R] 00000000 00000000		
00C290 _H	NEWDT22 [R] 00000000 00000000		NEWDT12 [R] 00000000 00000000		
00C294 _H	NEWDT42 [R] 00000000 00000000		NEWDT32 [R] 00000000 00000000		
00C298 _H	NEWDT62 [R] 00000000 00000000		NEWDT52 [R] 00000000 00000000		
00C29C _H	NEWDT82 [R] 00000000 00000000		NEWDT72 [R] 00000000 00000000		
00C2A0 _H	INTPND22 [R] 00000000 00000000		INTPND12 [R] 00000000 00000000		
00C2A4 _H	INTPND42 [R] 00000000 00000000		INTPND32 [R] 00000000 00000000		
00C2A8 _H	INTPND62 [R] 00000000 00000000		INTPND52 [R] 00000000 00000000		
00C2AC _H	INTPND82 [R] 00000000 00000000		INTPND72 [R] 00000000 00000000		
00C2B0 _H	MSGVAL22 [R] 00000000 00000000		MSGVAL12 [R] 00000000 00000000		
00C2B4 _H	MSGVAL42 [R] 00000000 00000000		MSGVAL32 [R] 00000000 00000000		
00C2B8 _H	MSGVAL62 [R] 00000000 00000000		MSGVAL52 [R] 00000000 00000000		
00C2BC _H	MSGVAL82 [R] 00000000 00000000		MSGVAL72 [R] 00000000 00000000		
00C2C0 _H - 00C2FC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C300 _H	CTRLR3 [R/W] 00000000 00000001		STATR3 [R/W] 00000000 00000000		CAN 3 Control Register
00C304 _H	ERRCNT3 [R] 00000000 00000000		BTR3 [R/W] 00100011 00000001		
00C308 _H	INTR3 [R] 00000000 00000000		TESTR3 [R/W] 00000000 X0000000		
00C30C _H	BRPE3 [R/W] 00000000 00000000		reserved		
00C310 _H	IF1CREQ3 [R/W] 00000000 00000001		IF1CMSK3 [R/W] 00000000 00000000		CAN 3 IF 1 Register
00C314 _H	IF1MSK23 [R/W] 11111111 11111111		IF1MSK13 [R/W] 11111111 11111111		
00C318 _H	IF1ARB23 [R/W] 00000000 00000000		IF1ARB13 [R/W] 00000000 00000000		
00C31C _H	IF1MCTR3 [R/W] 00000000 00000000		reserved		
00C320 _H	IF1DTA13 [R/W] 00000000 00000000		IF1DTA23 [R/W] 00000000 00000000		
00C324 _H	IF1DTB13 [R/W] 00000000 00000000		IF1DTB23 [R/W] 00000000 00000000		
00C328 _H	reserved				
00C32C _H	reserved				
00C330 _H	IF1DTA23 [R/W] 00000000 00000000		IF1DTA13 [R/W] 00000000 00000000		CAN 3 IF 1 Register mirror
00C334 _H	IF1DTB23 [R/W] 00000000 00000000		IF1DTB13 [R/W] 00000000 00000000		
00C338 _H	reserved				
00C33C _H	reserved				
00C340 _H	IF2CREQ3 [R/W] 00000000 00000001		IF2CMSK3 [R/W] 00000000 00000000		CAN 3 IF 2 Register
00C344 _H	IF2MSK23 [R/W] 11111111 11111111		IF2MSK13 [R/W] 11111111 11111111		
00C348 _H	IF2ARB23 [R/W] 00000000 00000000		IF2ARB13 [R/W] 00000000 00000000		
00C34C _H	IF2MCTR3 [R/W] 00000000 00000000		reserved		
00C350 _H	IF2DTA13 [R/W] 00000000 00000000		IF2DTA23 [R/W] 00000000 00000000		
00C354 _H	IF2DTB13 [R/W] 00000000 00000000		IF2DTB23 [R/W] 00000000 00000000		
00C358 _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C35C _H	reserved				
00C360 _H	IF2DTA23 [R/W] 00000000 00000000		IF2DTA13 [R/W] 00000000 00000000		CAN 3 IF 2 Register mirror
00C364 _H	IF2DTB23 [R/W] 00000000 00000000		IF2DTB13 [R/W] 00000000 00000000		
00C368 _H - 00C37C _H	reserved				
00C380 _H	TREQR23 [R] 00000000 00000000		TREQR13 [R] 00000000 00000000		CAN 3 Status Flags
00C384 _H	TREQR43 [R] 00000000 00000000		TREQR33 [R] 00000000 00000000		
00C388 _H	TREQR63 [R] 00000000 00000000		TREQR53 [R] 00000000 00000000		
00C38C _H	TREQR83 [R] 00000000 00000000		TREQR73 [R] 00000000 00000000		
00C390 _H	NEWDT23 [R] 00000000 00000000		NEWDT13 [R] 00000000 00000000		
00C394 _H	NEWDT43 [R] 00000000 00000000		NEWDT33 [R] 00000000 00000000		
00C398 _H	NEWDT63 [R] 00000000 00000000		NEWDT53 [R] 00000000 00000000		
00C39C _H	NEWDT83 [R] 00000000 00000000		NEWDT73 [R] 00000000 00000000		
00C3A0 _H	INTPND23 [R] 00000000 00000000		INTPND13 [R] 00000000 00000000		
00C3A4 _H	INTPND43 [R] 00000000 00000000		INTPND33 [R] 00000000 00000000		
00C3A8 _H	INTPND63 [R] 00000000 00000000		INTPND53 [R] 00000000 00000000		
00C3AC _H	INTPND83 [R] 00000000 00000000		INTPND73 [R] 00000000 00000000		
00C3B0 _H	MSGVAL23 [R] 00000000 00000000		MSGVAL13 [R] 00000000 00000000		
00C3B4 _H	MSGVAL43 [R] 00000000 00000000		MSGVAL33 [R] 00000000 00000000		
00C3B8 _H	MSGVAL63 [R] 00000000 00000000		MSGVAL53 [R] 00000000 00000000		
00C3BC _H	MSGVAL83 [R] 00000000 00000000		MSGVAL73 [R] 00000000 00000000		

Address	Register				Block
	+0	+1	+2	+3	
00C3C0 _H - 00C3FC _H	reserved				
00C400 _H	CTRLR4 [R/W] 00000000 00000001		STATR4 [R/W] 00000000 00000000		CAN 4 Control Register
00C404 _H	ERRCNT4 [R] 00000000 00000000		BTR4 [R/W] 00100011 00000001		
00C408 _H	INTR4 [R] 00000000 00000000		TESTR4 [R/W] 00000000 X0000000		
00C40C _H	BRPE4 [R/W] 00000000 00000000		reserved		
00C410 _H	IF1CREQ4 [R/W] 00000000 00000001		IF1CMSK4 [R/W] 00000000 00000000		CAN 4 IF 1 Register
00C414 _H	IF1MSK24 [R/W] 11111111 11111111		IF1MSK14 [R/W] 11111111 11111111		
00C418 _H	IF1ARB24 [R/W] 00000000 00000000		IF1ARB14 [R/W] 00000000 00000000		
00C41C _H	IF1MCTR4 [R/W] 00000000 00000000		reserved		
00C420 _H	IF1DTA14 [R/W] 00000000 00000000		IF1DTA24 [R/W] 00000000 00000000		
00C424 _H	IF1DTB14 [R/W] 00000000 00000000		IF1DTB24 [R/W] 00000000 00000000		
00C428 _H	reserved				
00C42C _H	reserved				
00C430 _H	IF1DTA24 [R/W] 00000000 00000000		IF1DTA14 [R/W] 00000000 00000000		CAN 4 IF 1 Register mirror
00C434 _H	IF1DTB24 [R/W] 00000000 00000000		IF1DTB14 [R/W] 00000000 00000000		
00C438 _H	reserved				
00C43C _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C440 _H	IF2CREQ4 [R/W] 00000000 00000001		IF2CMSK4 [R/W] 00000000 00000000		CAN 4 IF 2 Register
00C444 _H	IF2MSK24 [R/W] 11111111 11111111		IF2MSK14 [R/W] 11111111 11111111		
00C448 _H	IF2ARB24 [R/W] 00000000 00000000		IF2ARB14 [R/W] 00000000 00000000		
00C44C _H	IF2MCTR4 [R/W] 00000000 00000000		reserved		
00C450 _H	IF2DTA14 [R/W] 00000000 00000000		IF2DTA24 [R/W] 00000000 00000000		
00C454 _H	IF2DTB14 [R/W] 00000000 00000000		IF2DTB24 [R/W] 00000000 00000000		
00C458 _H	reserved				
00C45C _H	reserved				
00C460 _H	IF2DTA24 [R/W] 00000000 00000000		IF2DTA14 [R/W] 00000000 00000000		CAN 4 IF 2 Register mirror
00C464 _H	IF2DTB24 [R/W] 00000000 00000000		IF2DTB14 [R/W] 00000000 00000000		
00C468 _H - 00C47C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C480 _H	TREQR24 [R] 00000000 00000000		TREQR14 [R] 00000000 00000000		CAN 4 Status Flags
00C484 _H	TREQR44 [R] 00000000 00000000		TREQR34 [R] 00000000 00000000		
00C488 _H	TREQR64 [R] 00000000 00000000		TREQR54 [R] 00000000 00000000		
00C48C _H	TREQR84 [R] 00000000 00000000		TREQR74 [R] 00000000 00000000		
00C490 _H	NEWDT24 [R] 00000000 00000000		NEWDT14 [R] 00000000 00000000		
00C494 _H	NEWDT44 [R] 00000000 00000000		NEWDT34 [R] 00000000 00000000		
00C498 _H	NEWDT64 [R] 00000000 00000000		NEWDT54 [R] 00000000 00000000		
00C49C _H	NEWDT84 [R] 00000000 00000000		NEWDT74 [R] 00000000 00000000		
00C4A0 _H	INTPND24 [R] 00000000 00000000		INTPND14 [R] 00000000 00000000		
00C4A4 _H	INTPND44 [R] 00000000 00000000		INTPND34 [R] 00000000 00000000		
00C4A8 _H	INTPND64 [R] 00000000 00000000		INTPND54 [R] 00000000 00000000		
00C4AC _H	INTPND84 [R] 00000000 00000000		INTPND74 [R] 00000000 00000000		
00C4B0 _H	MSGVAL24 [R] 00000000 00000000		MSGVAL14 [R] 00000000 00000000		
00C4B4 _H	MSGVAL44 [R] 00000000 00000000		MSGVAL34 [R] 00000000 00000000		
00C4B8 _H	MSGVAL64 [R] 00000000 00000000		MSGVAL54 [R] 00000000 00000000		
00C4BC _H	MSGVAL84 [R] 00000000 00000000		MSGVAL74 [R] 00000000 00000000		
00C4C0 _H - 00C4FC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C500 _H	CTRLR5 [R/W] 00000000 00000001		STATR5 [R/W] 00000000 00000000		CAN 5 Control Register
00C504 _H	ERRCNT5 [R] 00000000 00000000		BTR5 [R/W] 00100011 00000001		
00C508 _H	INTR5 [R] 00000000 00000000		TESTR5 [R/W] 00000000 X0000000		
00C50C _H	BRPE5 [R/W] 00000000 00000000		reserved		
00C510 _H	IF1CREQ5 [R/W] 00000000 00000001		IF1CMSK5 [R/W] 00000000 00000000		CAN 5 IF 1 Register
00C514 _H	IF1MSK25 [R/W] 11111111 11111111		IF1MSK15 [R/W] 11111111 11111111		
00C518 _H	IF1ARB25 [R/W] 00000000 00000000		IF1ARB15 [R/W] 00000000 00000000		
00C51C _H	IF1MCTR5 [R/W] 00000000 00000000		reserved		
00C520 _H	IF1DTA15 [R/W] 00000000 00000000		IF1DTA25 [R/W] 00000000 00000000		
00C524 _H	IF1DTB15 [R/W] 00000000 00000000		IF1DTB25 [R/W] 00000000 00000000		
00C528 _H	reserved				
00C52C _H	reserved				
00C530 _H	IF1DTA25 [R/W] 00000000 00000000		IF1DTA15 [R/W] 00000000 00000000		CAN 5 IF 1 Register mirror
00C534 _H	IF1DTB25 [R/W] 00000000 00000000		IF1DTB15 [R/W] 00000000 00000000		
00C538 _H	reserved				
00C53C _H	reserved				
00C540 _H	IF2CREQ5 [R/W] 00000000 00000001		IF2CMSK5 [R/W] 00000000 00000000		CAN 5 IF 2 Register
00C544 _H	IF2MSK25 [R/W] 11111111 11111111		IF2MSK15 [R/W] 11111111 11111111		
00C548 _H	IF2ARB25 [R/W] 00000000 00000000		IF2ARB15 [R/W] 00000000 00000000		
00C54C _H	IF2MCTR5 [R/W] 00000000 00000000		reserved		
00C550 _H	IF2DTA15 [R/W] 00000000 00000000		IF2DTA25 [R/W] 00000000 00000000		
00C554 _H	IF2DTB15 [R/W] 00000000 00000000		IF2DTB25 [R/W] 00000000 00000000		
00C558 _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00C55C _H	reserved				
00C560 _H	IF2DTA25 [R/W] 00000000 00000000		IF2DTA15 [R/W] 00000000 00000000		CAN 5 IF 2 Register mirror
00C564 _H	IF2DTB25 [R/W] 00000000 00000000		IF2DTB15 [R/W] 00000000 00000000		
00C568 _H - 00C57C _H	reserved				
00C580 _H	TREQR25 [R] 00000000 00000000		TREQR15 [R] 00000000 00000000		CAN 5 Status Flags
00C584 _H	TREQR45 [R] 00000000 00000000		TREQR35 [R] 00000000 00000000		
00C588 _H	TREQR65 [R] 00000000 00000000		TREQR55 [R] 00000000 00000000		
00C58C _H	TREQR85 [R] 00000000 00000000		TREQR75 [R] 00000000 00000000		
00C590 _H	NEWDT25 [R] 00000000 00000000		NEWDT15 [R] 00000000 00000000		
00C594 _H	NEWDT45 [R] 00000000 00000000		NEWDT35 [R] 00000000 00000000		
00C598 _H	NEWDT65 [R] 00000000 00000000		NEWDT55 [R] 00000000 00000000		
00C59C _H	NEWDT85 [R] 00000000 00000000		NEWDT75 [R] 00000000 00000000		
00C5A0 _H	INTPND25 [R] 00000000 00000000		INTPND15 [R] 00000000 00000000		
00C5A4 _H	INTPND45 [R] 00000000 00000000		INTPND35 [R] 00000000 00000000		
00C5A8 _H	INTPND65 [R] 00000000 00000000		INTPND55 [R] 00000000 00000000		
00C5AC _H	INTPND85 [R] 00000000 00000000		INTPND75 [R] 00000000 00000000		
00C5B0 _H	MSGVAL25 [R] 00000000 00000000		MSGVAL15 [R] 00000000 00000000		
00C5B4 _H	MSGVAL45 [R] 00000000 00000000		MSGVAL35 [R] 00000000 00000000		
00C5B8 _H	MSGVAL65 [R] 00000000 00000000		MSGVAL55 [R] 00000000 00000000		
00C5BC _H	MSGVAL85 [R] 00000000 00000000		MSGVAL75 [R] 00000000 00000000		

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00C5C0 _H - 00CFFC _H	reserved				
00D000 _H	CIF0 [R] 00000100 11111111 01011011 11111111				FlexRay CIF
00D004 _H	CIF1 [R/W] 00000000 00000000 00000000 00000000				
00D008 _H	reserved				
00D00C _H	reserved				
00D010 _H	TEST1 [R/W] 00000000 00000000 00000011 00000000				FlexRay GIF
00D014 _H	TEST2 [R/W] 00000000 00000000 00000000 00000000				
00D018 _H	reserved				
00D01C _H	LCK [R/W] 00000000 00000000 00000000 00000000				

Address	Register				Block
	+0	+1	+2	+3	
00D020 _H	EIR [R/W] 00000000 00000000 00000000 00000000				FlexRay INT
00D024 _H	SIR [R/W] 00000000 00000000 00000000 00000000				
00D028 _H	EILS [R/W] 00000000 00000000 00000000 00000000				
00D02C _H	SILS [R/W] 00000011 00000011 11111111 11111111				
00D030 _H	EIES [R/W] 00000000 00000000 00000000 00000000				
00D034 _H	EIER [R/W] 00000000 00000000 00000000 00000000				
00D038 _H	SIES [R/W] 00000000 00000000 00000000 00000000				
00D03C _H	SIER [R/W] 00000000 00000000 00000000 00000000				
00D040 _H	ILE [R/W] 00000000 00000000 00000000 00000000				
00D044 _H	T0C [R/W] 00000000 00000000 00000000 00000000				
00D048 _H	T1C [R/W] 00000000 00000010 00000000 00000000				
00D04C _H	STPW1 [R/W] 00000000 00000000 00000000 00000000				
00D050 _H	STPW2 [R/W] 00000000 00000000 00000000 00000000				
00D050 _H - 00D07C _H	reserved				
00D080 _H	SUCC1 [R/W] 00001100 01000000 00010000 00000000				FlexRay SUC
00D084 _H	SUCC2 [R/W] 00000001 00000000 00000101 00000100				
00D088 _H	SUCC3 [R/W] 00000000 00000000 00000000 00010001				
00D08C _H	NEMC [R/W] 00000000 00000000 00000000 00000000				FlexRay NEM
00D090 _H	PRTC1 [R/W] 00001000 01001100 00000110 00110011				FlexRay PRT
00D094 _H	PRTC2 [R/W] 00001111 00101101 00001010 00001110				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00D098 _H	MHDC [R/W] 00000000 00000000 00000000 00000000				FlexRay MHD
00D09C _H	reserved				
00D0A0 _H	GTUC1 [R/W] 00000000 00000000 00000010 10000000				FlexRay GTU
00D0A4 _H	GTUC2 [R/W] 00000000 00000010 00000000 00001010				
00D0A8 _H	GTUC3 [R/W] 00000010 00000010 00000000 00000000				
00D0AC _H	GTUC4 [R/W] 00000000 00001000 00000000 00000111				
00D0B0 _H	GTUC5 [R/W] 00001110 00000000 00000000 00000000				
00D0B4 _H	GTUC6 [R/W] 00000000 00000010 00000000 00000000				
00D0B8 _H	GTUC7 [R/W] 00000000 00000010 00000000 00000100				
00D0BC _H	GTUC8 [R/W] 00000000 00000000 00000000 00000010				
00D0C0 _H	GTUC9 [R/W] 00000000 00000000 00000001 00000001				
00D0C4 _H	GTUC10 [R/W] 00000000 00000010 00000000 00000101				
00D0C8 _H	GTUC11 [R/W] 00000000 00000000 00000000 00000000				
00D0CC _H - 00D0FC _H	reserved				
00D100 _H	CCSV [R] 00000000 00010000 01000000 00000000				FlexRay SUC
00D104 _H	CCEV [R] 00000000 00000000 00000000 00000000				
00D108 _H - 00D10C _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00D110 _H	SCV [R] 00000000 00000000 00000000 00000000				FlexRay GTU
00D114 _H	MTCCV [R] 00000000 00000000 00000000 00000000				
00D118 _H	RCV [R] 00000000 00000000 00000000 00000000				
00D11C _H	OCV [R] 00000000 00000000 00000000 00000000				
00D120 _H	SFS [R] 00000000 00000000 00000000 00000000				
00D124 _H	SWNIT [R] 00000000 00000000 00000000 00000000				
00D128 _H	ACS [R/W] 00000000 00000000 00000000 00000000				
00D12C _H	reserved				
00D130 _H - 00D168 _H	ESIDn[1-15] [R] 00000000 00000000 00000000 00000000				
00D16C _H	reserved				
00D170 _H - 00D1A8 _H	OSIDn[1-15] [R] 00000000 00000000 00000000 00000000				
00D1AC _H	reserved				
00D1B0 _H - 00D1B8 _H	NMVn[1-3] [R] 00000000 00000000 00000000 00000000				FlexRay NEM
00D1BC _H - 00D2FC _H	reserved				
00D300 _H	MRC [R/W] 00000001 10000000 00000000 00000000				FlexRay MHD
00D304 _H	FRF [R/W] 00000001 10000000 00000000 00000000				
00D308 _H	FRFM [R/W] 00000000 00000000 00000000 00000000				
00D30C _H	FCL [R/W] 00000000 00000000 00000000 10000000				
00D310 _H	MHDS [R/W] 00000000 00000000 00000000 10000000				
00D314 _H	LDTS [R] 00000000 00000000 00000000 00000000				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00D318 _H	FSR [R] 00000000 00000000 00000000 00000000				FlexRay MHD
00D31C _H	MHDF [R/W] 00000000 00000000 00000000 00000000				
00D320 _H	TXRQ1 [R] 00000000 00000000 00000000 00000000				
00D324 _H	TXRQ2 [R] 00000000 00000000 00000000 00000000				
00D328 _H	TXRQ3 [R] 00000000 00000000 00000000 00000000				
00D32C _H	TXRQ4 [R] 00000000 00000000 00000000 00000000				
00D330 _H	NDAT1 [R] 00000000 00000000 00000000 00000000				FlexRay MHD
00D334 _H	NDAT2 [R] 00000000 00000000 00000000 00000000				
00D338 _H	NDAT3 [R] 00000000 00000000 00000000 00000000				
00D33C _H	NDAT4 [R] 00000000 00000000 00000000 00000000				
00D340 _H	MBSC1 [R] 00000000 00000000 00000000 00000000				
00D344 _H	MBSC2 [R] 00000000 00000000 00000000 00000000				
00D348 _H	MBSC3 [R] 00000000 00000000 00000000 00000000				
00D34C _H	MBSC4 [R] 00000000 00000000 00000000 00000000				
00D350 _H - 00D3EC _H	reserved				
00D3F0 _H	CREL [R] 00010000 00100110 00010000 00110001 ¹⁴				FlexRay GIF
00D3F4 _H	ENDN [R] 10000111 01100101 0100011 00100001				
00D3F8 _H - 00D3FC _H	reserved				

Address	Register				Block
	+0	+1	+2	+3	
00D400 _H - 00D4FC _H	WRDSn[1-64] [R/W] 00000000 00000000 00000000 00000000				FlexRay IBF
00D500 _H	WRHS1 [R/W] 00000000 00000000 00000000 00000000				
00D504 _H	WRHS2 [R/W] 00000000 00000000 00000000 00000000				
00D508 _H	WRHS3 [R/W] 00000000 00000000 00000000 00000000				
00D50C _H	reserved				
00D510 _H	IBCM [R/W] 00000000 00000000 00000000 00000000				
00D514 _H	IBCR [R/W] 00000000 00000000 00000000 00000000				
00D518 _H - 00D5FC _H	reserved				
00D600 _H - 00D6FC _H	RDDSn[1-64] [R] 00000000 00000000 00000000 00000000				FlexRay OBF
00D700 _H	RDHS1 [R] 00000000 00000000 00000000 00000000				
00D704 _H	RDHS2 [R] 00000000 00000000 00000000 00000000				
00D708 _H	RDHS3 [R] 00000000 00000000 00000000 00000000				
00D70C _H	MBS [R] 00000000 00000000 00000000 00000000				
00D710 _H	OBCM [R/W] 00000000 00000000 00000000 00000000				
00D714 _H	OBCR [R/W] 00000000 00000000 00000000 00000000				
00D718 _H - 00D7FC _H	reserved				
00D800 _H - 00EFFC _H	reserved				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00F000 _H	BCTRL [R/W] ----- 11111100 00000000				EDSU / MPU Control + IRQ
00F004 _H	BSTAT [R/W] ----- 000 00000000 10 - - 0000				
00F008 _H	BIAC [R] 00000000 00000000 00000000 00000000				
00F00C _H	BOAC [R] 00000000 00000000 00000000 00000000				
00F010 _H	BIRQ [R/W] 00000000 00000000 00000000 00000000				
00F014 _H - 00F01C _H	reserved				
00F020 _H	BCR0 [R/W] ----- 00000000 00000000 00000000				EDSU / MPU Control
00F024 _H	BCR1 [R/W] ----- 00000000 00000000 00000000				
00F028 _H	BCR2 [R/W] ----- 00000000 00000000 00000000				
00F02C _H	BCR3 [R/W] ----- 00000000 00000000 00000000				
00F030 _H	BCR4 [R/W] ----- 00000000 00000000 00000000				
00F034 _H	BCR5 [R/W] ----- 00000000 00000000 00000000				
00F038 _H	BCR6 [R/W] ----- 00000000 00000000 00000000				
00F03C _H	BCR7 [R/W] ----- 00000000 00000000 00000000				
00F040 _H - 00F07C _H	reserved				
00F080 _H	BAD0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 0
00F084 _H	BAD1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F088 _H	BAD2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F08C _H	BAD3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
00F090 _H	BAD4 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 1
00F094 _H	BAD5 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F098 _H	BAD6 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F09C _H	BAD7 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0A0 _H	BAD8 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 2
00F0A4 _H	BAD9 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0A8 _H	BAD10 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0AC _H	BAD11 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0B0 _H	BAD12 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 3
00F0B4 _H	BAD13 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0B8 _H	BAD14 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0BC _H	BAD15 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0C0 _H	BAD16 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 4
00F0C4 _H	BAD17 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0C8 _H	BAD18 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0CC _H	BAD19 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0D0 _H	BAD20 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 5
00F0D4 _H	BAD21 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0D8 _H	BAD22 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0DC _H	BAD23 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

MB91460 Series

Address	Register				Block
	+0	+1	+2	+3	
00F0E0 _H	BAD24 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 6
00F0E4 _H	BAD25 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0E8 _H	BAD26 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0EC _H	BAD27 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0F0 _H	BAD28 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				EDSU / MPU ch. 7
00F0F4 _H	BAD29 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0F8 _H	BAD30 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F0FC _H	BAD31 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
00F100 _H - 00FFFC _H	reserved				
010000 _H - 013FFC _H	Cache TAG way 1 (010000H - 0107FCH)				2 Way Set Associative I-Cache 4 KB
014000 _H - 017FFC _H	Cache TAG way 2 (014000H - 0147FCH)				
018000 _H - 01BFFC _H	Cache RAM way 1 (018000H - 0187FCH)				
01C000 _H - 01FFFC _H	Cache RAM way 2 01C000H - 01C7FCH)				
020000 _H - 02FFFC _H	MB91V460A, MB91FV460B D-RAM size is 64 KB (data access is 0 waitcycles)				Data-RAM max. 64 KB
030000 _H - 03FFFC _H	MB91V460A, MB91FV460B I-/D-RAM if size is 64 KB (instruction access is 0 waitcycles, data access is 1 waitcycle)				Instruction/ Data RAM max. 64 KB
040000 _H - 05FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS00 *15 area (128 KB)
060000 _H - 07FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS01 area (128 KB)

Address	Register				Block
	+0	+1	+2	+3	
080000 _H - 09FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS02 area (128 KB)
0A0000 _H - 0BFFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS03 area (128 KB)
0C0000 _H - 0DFFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS04 area (128 KB)
0E0000 _H - 0FFFF4 _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS05 area (128 KB)
0FFFF8 _H	FMV [R] ¹⁶ 06 00 00 00H				Fixed Mode Vector
0FFFFC _H	FRV [R] ¹⁷ 00 00 BF F8H				Fixed Reset Vector
100000 _H - 13FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS06 area (256 KB)
140000 _H - 17FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS07 area (256 KB)
180000 _H - 1BFFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS08 area (256 KB)
1C0000 _H - 1FFFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS09 area (256 KB)
200000 _H - 27FFFC _H	Internal Flash memory or external evaluation S-RAM or external bus area				ROMS10 area (512 KB)
280000 _H - 2FFFFC _H	External evaluation S-RAM or external bus area				ROMS11 area (512 KB)
300000 _H - 37FFFC _H					ROMS12 area (512 KB)
380000 _H - 3FFFFC _H					ROMS13 area (512 KB)
400000 _H - 47FFFC _H					ROMS14 area (512 KB)

Address	Register				Block
	+0	+1	+2	+3	
480000 _H - 4FFFFC _H					ROMS15 area (512 KB)
500000 _H - FFFBEFFC _H	External Bus area				External bus area
FFFBF000 _H - FFFCFFFC _H	Data Flash area (if enabled) or External Bus area, Data Flash on MB91FV460B is 64 KB + 256 Byte				Data Flash area
FFFD0000 _H - FFFFFFFC _H	External Bus area				External bus area

1. Use a read access (byte or halfword) to this address to synchronize the CPU operation (e.g. the interrupt acceptance of the CPU) to a preceding write access to the resources on R-bus (e.g. to an interrupt flag) on following addresses (0x0000-0x01FF, 0x0280-0x038B, 0x0394-0x03B7, 0x0400-0x063F, 0x0688-0x075F and 0x0C00-0x0FFF).
2. Write access to ROMS register is only possible on MB91V460A and MB91FV460B. The initial value of ROMS is different versus MB91V460A
3. Clock Monitor counter for test purposes. Write access clears the counter to 0x00.
4. Writable only once. PPMUX is reset by INIT and SoftReset.
5. Writable only once. PPMUX2 is reset by INIT and SoftReset.
6. Number of bits depends on the number of available CAN channels
7. ACRO[11:10] depends on bus width setting in Mode vector fetch information
8. TCR[3:0] INIT value = 0000, keeps value after RST
9. Always write 1 to IOS[1] !
10. Initial value of FMWT[7:0] and FMWT2 is different versus MB91V460A
11. APIX® is a registered mark of INOVA Semiconductors GmbH.
12. Datecode of APIX® module version
13. Use a read access (byte or halfword) to this address to synchronize the CPU operation (e.g. the interrupt acceptance of the CPU) to a preceding write access to the CANs on D-bus (e.g. to an interrupt flag) on following addresses (0xC000-0xFFFF).
14. Datecode of FlexRay Core release (revision 1.0.2 on MB91FV460B)
15. The ROMS register can be written by the CPU. Initial value is 0xF800, initialization by Power-On Reset.
16. If the control pin FIX_ENX is 0, then write operations to address 0FFFF8H are not possible. When reading this address, the values shown above will be read.
17. If the control pin FIX_ENX is 0, then write operations to address 0FFFFCH are not possible. When reading this address, the values shown above will be read.

MB91460 Series**3.3. Interrupt Vector Table**

This section shows the allocation of interrupt and interrupt vector/interrupt register.

Table 3.3-1 Interrupt Vector Table

Interrupt	Interrupt number		Interrupt level ^{*1}		Interrupt vector ^{*2}		DMA	
	Decimal	Hexadecimal	Setting Register	Register address	Offset	Default Vector address	RN ^{*8}	Stop ^{*9}
Reset	0	00	—	—	3FC _H	000FFFFC _H	—	—
Mode vector	1	01	—	—	3F8 _H	000FFFF8 _H	—	—
System reserved	2	02	—	—	3F4 _H	000FFFF4 _H	—	—
System reserved	3	03	—	—	3F0 _H	000FFFF0 _H	—	—
System reserved	4	04	—	—	3EC _H	000FFFE _C	—	—
CPU supervisor mode (INT #5 instruction) ^{*6}	5	05	—	—	3E8 _H	000FFFE8 _H	—	—
Memory Protection exception ^{*6}	6	06	—	—	3E4 _H	000FFFE4 _H	—	—
Co-processor fault trap ^{*5}	7	07	—	—	3E0 _H	000FFFE0 _H	—	—
Co-processor error trap ^{*5}	8	08	—	—	3DC _H	000FFFD _C	—	—
INTE instruction ^{*5}	9	09	—	—	3D8 _H	000FFFD8 _H	—	—
Instruction break exception ^{*5}	10	0A	—	—	3D4 _H	000FFFD4 _H	—	—
Operand break trap ^{*5}	11	0B	—	—	3D0 _H	000FFFD0 _H	—	—
Step trace trap ^{*5}	12	0C	—	—	3CC _H	000FFFC _C	—	—
NMI interrupt (DSU4) ^{*5}	13	0D	—	—	3C8 _H	000FFFC8 _H	—	—
Undefined instruction exception	14	0E	—	—	3C4 _H	000FFFC4 _H	—	—
NMI request	15	0F	F _H fixed		3C0 _H	000FFFC0 _H	—	—
External Interrupt 0 External Interrupt 16	16	10	ICR00	440 _H	3BC _H	000FFFB _C	0, 16 136	—
External Interrupt 1 External Interrupt 17	17	11			3B8 _H	000FFFB8 _H	1, 17 137	—
External Interrupt 2 External Interrupt 18	18	12	ICR01	441 _H	3B4 _H	000FFFB4 _H	2, 18 138	—
External Interrupt 3 External Interrupt 19	19	13			3B0 _H	000FFFB0 _H	3, 19 139	—
External Interrupt 4 External Interrupt 20	20	14	ICR02	442 _H	3AC _H	000FFFA _C	20 140	—
External Interrupt 5 External Interrupt 21	21	15			3A8 _H	000FFFA8 _H	21 141	—

Interrupt	Interrupt number		Interrupt level *1		Interrupt vector *2		DMA	
	Deci-mal	Hexa-deci-mal	Setting Register	Register address	Offset	Default Vector address	RN *8	Stop *9
External Interrupt 6 External Interrupt 22	22	16	ICR03	443 _H	3A4 _H	000FFFA4 _H	22 142	—
External Interrupt 7 External Interrupt 23	23	17			3A0 _H	000FFFA0 _H	23 143	—
External Interrupt 8 External Interrupt 24	24	18	ICR04	444 _H	39C _H	000FFF9C _H	—	—
External Interrupt 9 External Interrupt 25	25	19			398 _H	000FFF98 _H	—	—
External Interrupt 10 External Interrupt 26	26	1A	ICR05	445 _H	394 _H	000FFF94 _H	—	—
External Interrupt 11 External Interrupt 27	27	1B			390 _H	000FFF90 _H	—	—
External Interrupt 12 External Interrupt 28	28	1C	ICR06	446 _H	38C _H	000FFF8C _H	—	—
External Interrupt 13 External Interrupt 29	29	1D			388 _H	000FFF88 _H	—	—
External Interrupt 14 External Interrupt 30	30	1E	ICR07	447 _H	384 _H	000FFF84 _H	—	—
External Interrupt 15 External Interrupt 31	31	1F			380 _H	000FFF80 _H	—	—
Reload Timer 0 Reload Timer 8	32	20	ICR08	448 _H	37C _H	000FFF7C _H	4, 32 128	—
Reload Timer 1 Reload Timer 9	33	21			378 _H	000FFF78 _H	5, 33 129	—
Reload Timer 2 Reload Timer 10	34	22	ICR09	449 _H	374 _H	000FFF74 _H	34 130	—
Reload Timer 3 Reload Timer 11	35	23			370 _H	000FFF70 _H	35 131	—
Reload Timer 4 Reload Timer 12	36	24	ICR10	44A _H	36C _H	000FFF6C _H	36 132	—
Reload Timer 5 Reload Timer 13	37	25			368 _H	000FFF68 _H	37 133	—
Reload Timer 6 Reload Timer 14	38	26	ICR11	44B _H	364 _H	000FFF64 _H	38 134	—
Reload Timer 7 Reload Timer 15	39	27			360 _H	000FFF60 _H	39 135	—
Free Run Timer 0 Free Run Timer 8	40	28	ICR12	44C _H	35C _H	000FFF5C _H	40 176	—
Free Run Timer 1 Free Run Timer 9	41	29			358 _H	000FFF58 _H	41 177	—

MB91460 Series

Interrupt	Interrupt number		Interrupt level ^{*1}		Interrupt vector ^{*2}		DMA	
	Deci- mal	Hexa- deci- mal	Setting Register	Register address	Offset	Default Vector address	RN ^{*8}	Stop ^{*9}
Free Run Timer 2 Free Run Timer 10	42	2A	ICR13	44D _H	354 _H	000FFF54 _H	42 178	—
Free Run Timer 3 Free Run Timer 11	43	2B			350 _H	000FFF50 _H	43 179	—
Free Run Timer 4	44	2C	ICR14	44E _H	34C _H	000FFF4C _H	44	—
Free Run Timer 5	45	2D			348 _H	000FFF48 _H	45	—
Free Run Timer 6	46	2E	ICR15	44F _H	344 _H	000FFF44 _H	46	—
Free Run Timer 7	47	2F			340 _H	000FFF40 _H	47	—
CAN 0	48	30	ICR16	450 _H	33C _H	000FFF3C _H	—	
CAN 1	49	31			338 _H	000FFF38 _H	—	
CAN 2	50	32	ICR17	451 _H	334 _H	000FFF34 _H	—	
CAN 3	51	33			330 _H	000FFF30 _H	—	
CAN 4	52	34	ICR18	452 _H	32C _H	000FFF2C _H	—	
CAN 5	53	35			328 _H	000FFF28 _H	—	
LIN-USART 0 RX	54	36	ICR19	453 _H	324 _H	000FFF24 _H	6, 48	6, 48
LIN-USART 0 TX LIN-USART 0 EOT	55	37			320 _H	000FFF20 _H	7, 49 ---	—
LIN-USART 1 RX	56	38	ICR20	454 _H	31C _H	000FFF1C _H	8, 50	8, 50
LIN-USART 1 TX LIN-USART 1 EOT	57	39			318 _H	000FFF18 _H	9, 51 ---	
LIN-USART 2 RX	58	3A	ICR21	455 _H	314 _H	000FFF14 _H	52	52
LIN-USART 2 TX LIN-USART 2 EOT	59	3B			310 _H	000FFF10 _H	53 ---	—
LIN-USART 3 RX	60	3C	ICR22	456 _H	30C _H	000FFF0C _H	54	54
LIN-USART 3 TX LIN-USART 3 EOT	61	3D			308 _H	000FFF08 _H	55 ---	—
System reserved	62	3E	ICR23 ^{*4}	457 _H	304 _H	000FFF04 _H	—	
Delayed Interrupt	63	3F			300 _H	000FFF00 _H	—	
System reserved ^{*3}	64	40	(ICR24)	(458 _H)	2FC _H	000FFEFC _H	—	
System reserved ^{*3}	65	41			2F8 _H	000FFE8 _H	—	
LIN-USART 4 RX	66	42	ICR25	459 _H	2F4 _H	000FFE4 _H	10, 56	10, 56
LIN-USART 4 TX LIN-USART 4 EOT	67	43			2F0 _H	000FFE0 _H	11, 57 ---	—
LIN-USART 5 RX	68	44	ICR26	45A _H	2EC _H	000FEEC _H	12, 58	12, 58
LIN-USART 5 TX LIN-USART 5 EOT	69	45			2E8 _H	000FEE8 _H	13, 59 ---	—

Interrupt	Interrupt number		Interrupt level *1		Interrupt vector *2		DMA	
	Deci-mal	Hexa-deci-mal	Setting Register	Register address	Offset	Default Vector address	RN *8	Stop *9
LIN-USART 6 RX	70	46	ICR27	45B _H	2E4 _H	000FFEE4 _H	60	60
LIN-USART 6 TX LIN-USART 6 EOT	71	47			2E0 _H	000FFEE0 _H	61 ---	
LIN-USART 7 RX	72	48	ICR28	45C _H	2DC _H	000FFEDC _H	62	62
LIN-USART 7 TX LIN-USART 7 EOT	73	49			2D8 _H	000FFED8 _H	63 ---	
I ² C 0 / I ² C 2	74	4A	ICR29	45D _H	2D4 _H	000FFED4 _H	28/30	28/30
I ² C 1 / I ² C 3	75	4B			2D0 _H	000FFED0 _H	29/31	29/31
LIN-USART 8 RX / APIX [®] Event / Fatal Error	76	4C	ICR30	45E _H	2CC _H	000FFECC _H	64 / ---	64 / ---
LIN-USART 8 TX / LIN-USART 8 EOT ADC 0 Range Comparator	77	4D			2C8 _H	000FFEC8 _H	65 / --- ---	--- ---
LIN-USART 9 RX / I2C 4 / I2C 6	78	4E	ICR31	45F _H	2C4 _H	000FFEC4 _H	66 / 188/190	66 / 188/190
LIN-USART 9 TX / LIN-USART 9 EOT I2C 5 / I2C 7	79	4F			2C0 _H	000FFEC0 _H	67 / --- 189/191	--- / --- 189/191
LIN-USART 10 RX / APIX [®] Transaction Buffer	80	50	ICR32	460 _H	2BC _H	000FFEBCH	68/ 160-175	68 / 160-175
LIN-USART 10 TX LIN-USART 10 EOT ADC 1 Range Comparator	81	51			2B8 _H	000FFEB8 _H	69 / --- ---	--- ---
LIN-USART 11 RX / Media LB FIFO Buffer	82	52	ICR33	461 _H	2B4 _H	000FFEB4 _H	70 / ---	70 / ---
LIN-USART 11 TX LIN-USART 11 EOT	83	53			2B0 _H	000FFEB0 _H	71 ---	---
LIN-USART 12 RX / FlexRay Channel 0	84	54	ICR34	462 _H	2AC _H	000FFEAC _H	72 / 116(IBF), 117(OBF)	72 / ---, ---
LIN-USART 12 TX / LIN-USART 12 EOT FlexRay Timer Channel 0	85	55			2A8 _H	000FFEA8 _H	73 / --- ---	--- / ---
LIN-USART 13 RX / FlexRay Channel 1	86	56	ICR35	463 _H	2A4 _H	000FFEA4 _H	74 / 118 (IBF), 119(OBF)	74 / ---, ---
LIN-USART 13 TX / LIN-USART 13 EOT FlexRay Timer Channel 1	87	57			2A0 _H	000FFEA0 _H	75 / --- ---	--- / ---

MB91460 Series

Interrupt	Interrupt number		Interrupt level *1		Interrupt vector *2		DMA	
	Deci-mal	Hexa-deci-mal	Setting Register	Register address	Offset	Default Vector address	RN *8	Stop *9
LIN-USART 14 RX / Media LB	88	58	ICR36	464 _H	29C _H	000FFE9C _H	76 / —	76 / —
LIN-USART 14 TX / LIN-USART 14 EOT I2S Channels 0-9 Error	89	59			298 _H	000FFE98 _H	77 / --- —	—
LIN-USART 15 RX / I2S Even Channels Tx/Rx	90	5A	ICR37	465 _H	294 _H	000FFE94 _H	78 / 125	—
LIN-USART 15 TX / LIN-USART 15 EOT I2S Odd Channels Tx/Rx	91	5B			290 _H	000FFE90 _H	79 / --- 126	—
Input Capture 0 Input Capture 8	92	5C	ICR38	466 _H	28C _H	000FFE8C _H	80 180	—
Input Capture 1 Input Capture 9	93	5D			288 _H	000FFE88 _H	81 181	—
Input Capture 2	94	5E	ICR39	467 _H	284 _H	000FFE84 _H	82	—
Input Capture 3	95	5F			280 _H	000FFE80 _H	83	—
Input Capture 4	96	60	ICR40	468 _H	27C _H	000FFE7C _H	84	—
Input Capture 5 USB Endpoint 0 IN (IRQ0)	97	61			278 _H	000FFE78 _H	85 200	—
Input Capture 6 USB Endpoint 0 OUT (IRQ1)	98	62	ICR41	469 _H	274 _H	000FFE74 _H	86 201	—
Input Capture 7 USB Endpoint 1 (IRQ2)	99	63			270 _H	000FFE70 _H	87 202	—
Output Compare 0 USB Endpoint 2 (IRQ3)	100	64	ICR42	46A _H	26C _H	000FFE6C _H	88 203	—
Output Compare 1 USB Endpoint 3 (IRQ4)	101	65			268 _H	000FFE68 _H	89 204	—
Output Compare 2 USB Endpoint 4 (IRQ5)	102	66	ICR43	46B _H	264 _H	000FFE64 _H	90 205	—
Output Compare 3 USB Endpoint 5 (IRQ6)	103	67			260 _H	000FFE60 _H	91 206	—
Output Compare 4 USB Func Flags1 (IRQ7) *10	104	68	ICR44	46C _H	25C _H	000FFE5C _H	92 ---	—
Output Compare 5 USB Func Flags2 (IRQ8) *11	105	69			258 _H	000FFE58 _H	93 ---	—
Output Compare 6 USB Host Flags1 (IRQ9) *12	106	6A	ICR45	46D _H	254 _H	000FFE54 _H	94 ---	—
Output Compare 7 USB Host Flags2 (IRQ10) *13	107	6B			250 _H	000FFE50 _H	95 ---	—

Interrupt	Interrupt number		Interrupt level *1		Interrupt vector *2		DMA	
	Deci-mal	Hexa-deci-mal	Setting Register	Register address	Offset	Default Vector address	RN *8	Stop *9
Sound Generator 0	108	6C	ICR46	46E _H	24C _H	000FFE4C _H	—	—
Phase Frequency Modulator	109	6D			248 _H	000FFE48 _H	114	—
System reserved	110	6E	ICR47 *4	46F _H	244 _H	000FFE44 _H	—	—
System reserved	111	6F			240 _H	000FFE40 _H	—	—
Prog. Pulse Generator 0 Prog. Pulse Generator 16	112	70	ICR48	470 _H	23C _H	000FFE3C _H	15, 96 144	—
Prog. Pulse Generator 1 Prog. Pulse Generator 17	113	71			238 _H	000FFE38 _H	97 145	—
Prog. Pulse Generator 2 Prog. Pulse Generator 18	114	72	ICR49	471 _H	234 _H	000FFE34 _H	98 146	—
Prog. Pulse Generator 3 Prog. Pulse Generator 19	115	73			230 _H	000FFE30 _H	99 147	—
Prog. Pulse Generator 4 Prog. Pulse Generator 20	116	74	ICR50	472 _H	22C _H	000FFE2C _H	100 148	—
Prog. Pulse Generator 5 Prog. Pulse Generator 21	117	75			228 _H	000FFE28 _H	101 149	—
Prog. Pulse Generator 6 Prog. Pulse Generator 22	118	76	ICR51	473 _H	224 _H	000FFE24 _H	102 150	—
Prog. Pulse Generator 7 Prog. Pulse Generator 23	119	77			220 _H	000FFE20 _H	103 151	—
Prog. Pulse Generator 8 Prog. Pulse Generator 24	120	78	ICR52	474 _H	21C _H	000FFE1C _H	104 152	—
Prog. Pulse Generator 9 Prog. Pulse Generator 25	121	79			218 _H	000FFE18 _H	105 153	—
Prog. Pulse Generator 10 Prog. Pulse Generator 26	122	7A	ICR53	475 _H	214 _H	000FFE14 _H	106 154	—
Prog. Pulse Generator 11 Prog. Pulse Generator 27	123	7B			210 _H	000FFE10 _H	107 155	—
Prog. Pulse Generator 12 Prog. Pulse Generator 28	124	7C	ICR54	476 _H	20C _H	000FFE0C _H	108 156	—
Prog. Pulse Generator 13 Prog. Pulse Generator 29	125	7D			208 _H	000FFE08 _H	109 157	—
Prog. Pulse Generator 14 Prog. Pulse Generator 30	126	7E	ICR55	477 _H	204 _H	000FFE04 _H	110 158	—
Prog. Pulse Generator 15 Prog. Pulse Generator 31	127	7F			200 _H	000FFE00 _H	111 159	—
Up/Down Counter 0	128	80	ICR56	478 _H	1FC _H	000FFDFC _H	24	—
Up/Down Counter 1	129	81			1F8 _H	000FFDF8 _H	25	—

MB91460 Series

3.3.Interrupt Vector Table

Interrupt	Interrupt number		Interrupt level ^{*1}		Interrupt vector ^{*2}		DMA	
	Deci-mal	Hexa-deci-mal	Setting Register	Register address	Offset	Default Vector address	RN ^{*8}	Stop ^{*9}
Up/Down Counter 2	130	82	ICR57	479 _H	1F4 _H	000FFDF4 _H	26	—
Up/Down Counter 3	131	83			1F0 _H	000FFDF0 _H	27	—
Real Time Clock	132	84	ICR58	47A _H	1EC _H	000FFDEC _H	—	—
Calibration Unit Programmable CRC Mod.	133	85			1E8 _H	000FFDE8 _H	— 199	—
A/D Converter 0 End of Conv A/D Converter 0 End of Scan	134	86	ICR59	47B _H	1E4 _H	000FFDE4 _H	14, 112 120	— —
A/D Converter 1 End of Conv A/D Converter 1 End of Scan	135	87			1E0 _H	000FFDE0 _H	113 121	— —
Alarm Comparator 0	136	88	ICR60	47C _H	1DC _H	000FFDDC _H	—	—
Alarm Comparator 1	137	89			1D8 _H	000FFDD8 _H	—	—
Low Voltage Detection	138	8A	ICR61	47D _H	1D4 _H	000FFDD4 _H	—	—
reserved	139	8B			1D0 _H	000FFDD0 _H	—	—
Timebase Overflow	140	8C	ICR62	47E _H	1CC _H	000FFDCC _H	—	—
PLL Clock Gear Data Flash Write Complete	141	8D			1C8 _H	000FFDC8 _H	— 195	— 195
DMA Controller	142	8E	ICR63	47F _H	1C4 _H	000FFDC4 _H	—	—
Main/Sub OSC stability wait	143	8F			1C0 _H	000FFDC0 _H	—	—
Boot Security vector ^{*7}	144	90	—	—	1BC _H	000FFDBC _H	—	—
Used by the INT instruction.	145 to 255	91 to FF	—	—	1B8 _H to 000 _H	000FFDB8 _H to 000FFC00 _H	—	—

^{*1} The The Interrupt Control Registers (ICRs) are located in the interrupt controller and set the interrupt level for each interrupt request. An ICR is provided for each interrupt request.

^{*2} The vector address for each EIT (exception, interrupt or trap) is calculated by adding the listed offset to the table base register value (TBR). The TBR specifies the top of the EIT vector table. The addresses listed in the table are for the default TBR value (0x000FFC00). The TBR is initialized to this value by a reset. After execution of the internal boot ROM TBR is set to 0x000FFC00.

^{*3} Used by REALOS

^{*4} ICR23 and ICR47 can be exchanged by setting the REALOS compatibility bit (addr 0x0C03 : IOS[0])

^{*5} System reserved

^{*6} Memory Protection Unit (MPU) support

^{*7} Only for MB91V460. Please see the hardware manual, chapter “Fixed Mode-Reset Vector” for boot security vectors used on flash devices.

^{*8} DMA RN is the resource number used for DMA operation. No number means that this resource interrupt cannot be used to trigger a DMA transfer.

^{*9} DMA Stop shows the DMA Transfer Stop Request feature.

^{*10} USB Function Flags 1 : SUSP, SOF, BRST, WKUP, CONF.

^{*11} USB Function Flags 2: SPK

^{*12} USB MiniHost Flags 1: DIRQ, CNNIRQ, URIRQ, RWKIRQ.

^{*13} USB MiniHost Flags 2 : SOFIRQ, CMPIRQ.



MB91460 Series

3.4. Package

3.4.1 MB91V460A: BGA-660P-M02 package

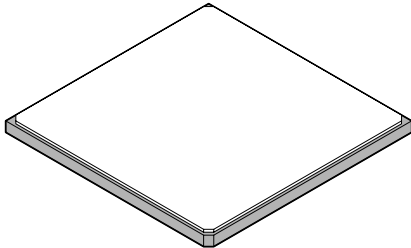
Figure 3.4-1 External Dimension of BGA-660P-M02

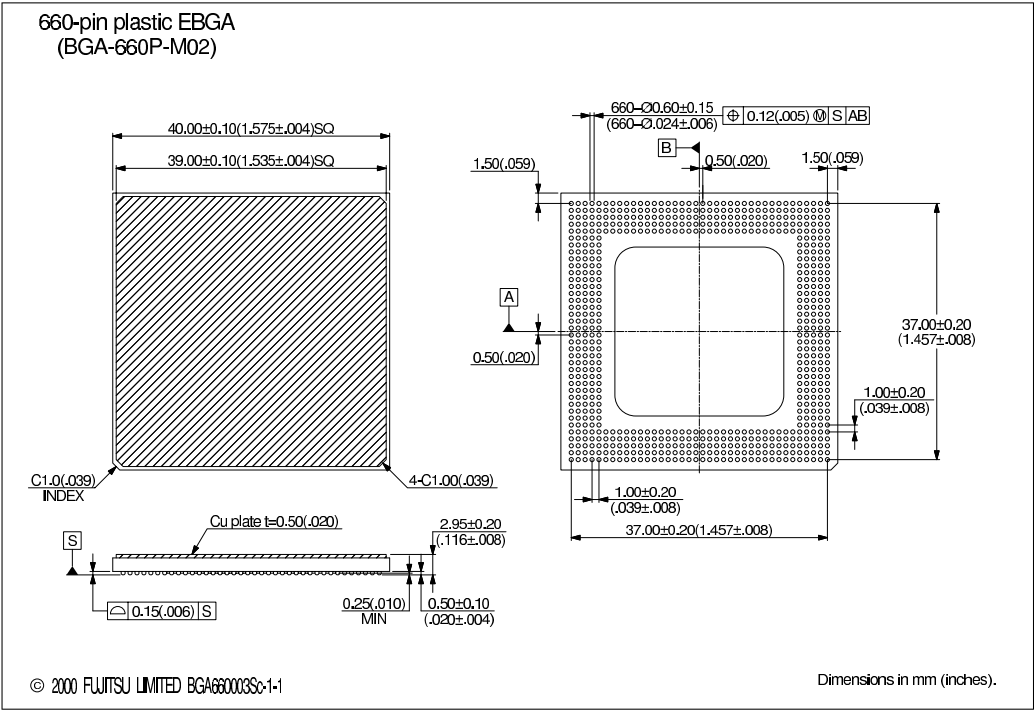
FUJITSU SEMICONDUCTOR
DATA SHEET

ENHANCED BALL GRID ARRAY PACKAGE
660 PIN PLASTIC

To Top / Package Lineup / Package Index

BGA-660P-M02

<div>660-pin plastic EBGA</div>  <div>(BGA-660P-M02)</div>	Ball pitch	1.00 mm
	Ball matrix	38 × 38
	Package width × package length	40.00 × 40.00 mm
	Sealing method	Resin sealed
	Ball size	Ø 0.6 mm
	Mounting height	3.15 mm MAX
	Weight	14.2g



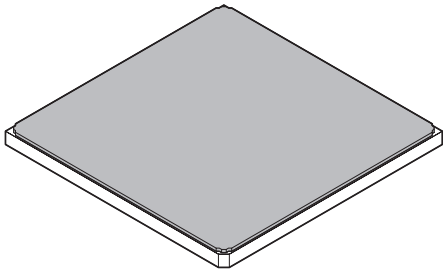
The contents of this document are subject to change without notice.
Customers are advised to consult with FUJITSU sales representatives before ordering.
FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of the information or package dimensions in this document.

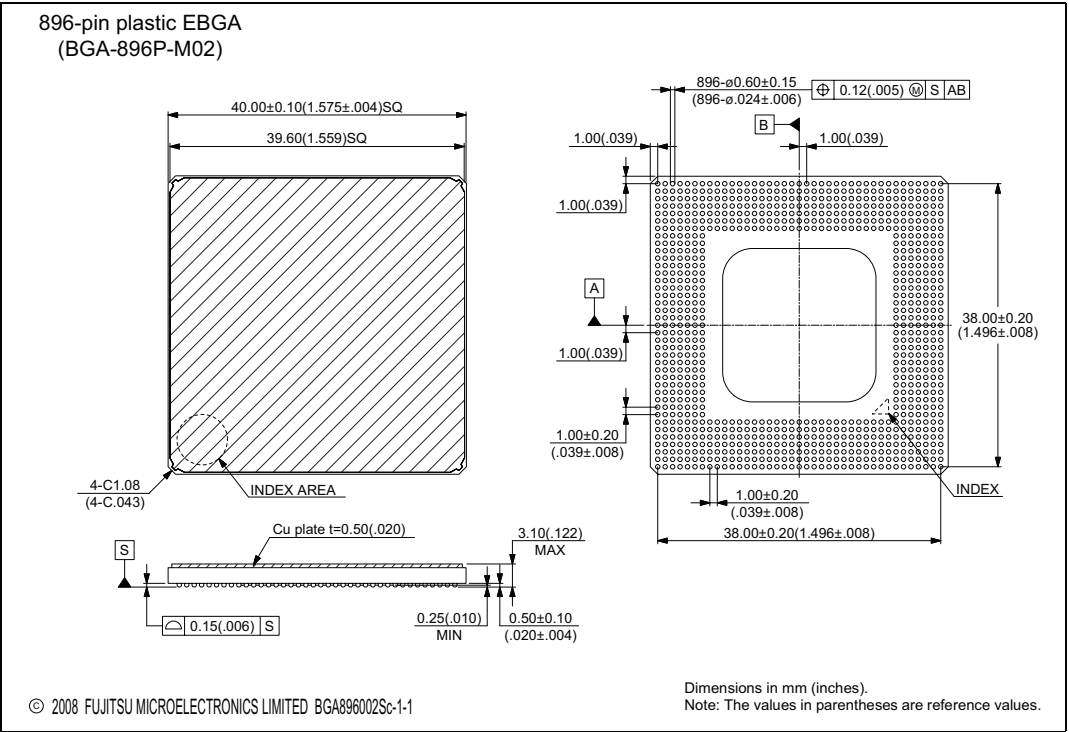
3.4.2 MB91FV460B: BGA-896P-M02 package

Figure 3.4-2 External Dimension of BGA-896P-M02
FUJITSU MICROELECTRONICS
DATA SHEET

ENHANCED BALL GRID ARRAY PACKAGE
896 PIN PLASTIC

BGA-896P-M02

<div>896-pin plastic EPGA</div> <div></div> <div>(BGA-896P-M02)</div>	Ball pitch	1.00 mm
	Package width × package length	40.00 × 40.00 mm
	Lead shape	Soldering ball
	Sealing method	Resin seal
	Ball size	0.60 mm
	Mounting height	3.10 mm MAX
	Weight	14.20 g



The contents of this document are subject to change without notice.
Customers are advised to consult with FUJITSU sales representatives before ordering.
FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of the information or package dimensions in this document.

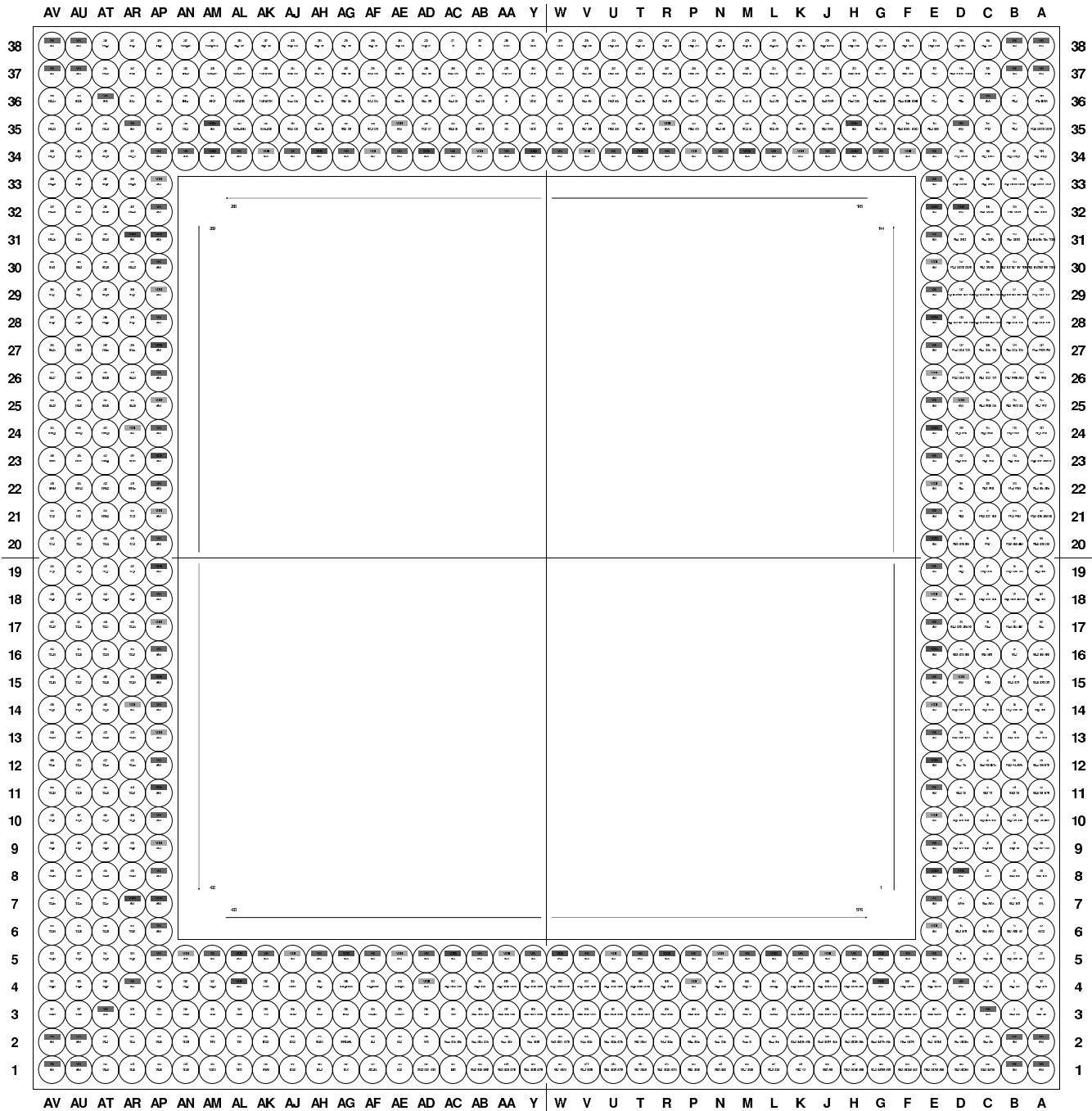
0307

MB91460 Series

3.5. Pin Assignment Diagram

3.5.1 MB91V460A (BG-660P-M02 package)

Figure 3.5-1 Pin Assignment Diagram of MB91V460A



3.5.2 MB91FV460B (BGA-896P-M02 package, top view)

Figure 3.5-2 Pin Assignment Diagram of MB91FV460B, top view

[illegible]

MB91460 Series

3.5.Pin Assignment Diagram

3.5.3 MB91FV460B (BGA-896P-M02 package, bottom view)

Figure 3.5-3 Pin Assignment Diagram of MB91FV460B, bottom view

▲		A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	
2	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	40	
3	151	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	190	41	
4	150	295	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	332	191	42	
5	149	294	431	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	466	333	192	43	
6	148	293	430	559	560	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	592	467	334	193	44	
7	147	292	429	558	679	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	710	593	468	335	194	45	
8	146	291	428	557	678	791	896																																	
9	145	290	427	556	677	790	895																																	
10	144	289	426	555	676	789	894																																	
11	143	288	425	554	675	788	893																																	
12	142	287	424	553	674	787	892																																	
13	141	286	423	552	673	786	891																																	
14	140	285	422	551	672	785	890																																	
15	139	284	421	550	671	784	889																																	
16	138	283	420	549	670	783	888																																	
17	137	282	419	548	669	782	887																																	
18	136	281	418	547	668	781	886																																	
19	135	280	417	546	667	780	885																																	
20	134	279	416	545	666	779	884																																	
21	133	278	415	544	665	778	883																																	
22	132	277	414	543	664	777	882																																	
23	131	276	413	542	663	776	881																																	
24	130	275	412	541	662	775	880																																	
25	129	274	411	540	661	774	879																																	
26	128	273	410	539	660	773	878																																	
27	127	272	409	538	659	772	877																																	
28	126	271	408	537	658	771	876																																	
29	125	270	407	536	657	770	875																																	
30	124	269	406	535	656	769	874																																	
31	123	268	405	534	655	768	873																																	
32	122	267	404	533	654	767	872																																	
33	121	266	403	532	653	766	871		870	869	868	867	866	865	864	863	862	861	860	859	858	857	856	855	854	853	852	851	850	849	848	847	846	845	736	619	494	361	220	71
34	120	265	402	531	652	765	870		763	762	761	760	759	758	757	756	755	754	753	752	751	750	749	748	747	746	745	744	743	742	741	740	739	738	737	620	495	362	221	72
35	119	264	401	530	651	650	649		648	647	646	645	644	643	642	641	640	639	638	637	636	635	634	633	632	631	630	629	628	627	626	625	624	623	622	621	496	363	222	73
36	118	263	400	529	528	527	526		525	524	523	522	521	520	519	518	517	516	515	514	513	512	511	510	509	508	507	506	505	504	503	502	501	500	499	498	497	364	223	74
37	117	262	399	398	397	396	395		394	393	392	391	390	389	388	387	386	385	384	383	382	381	380	379	378	377	376	375	374	373	372	371	370	369	368	367	366	365	224	75
38	116	261	260	259	258	257	256		255	254	253	252	251	250	249	248	247	246	245	244	243	242	241	240	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	76
39	115	114	113	112	111	110	109		108	107	106	105	104	103	102	101	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	

MB91FV460B
BGA896-08EK
Bottom View
(FJ pin)

3.6. Pin Definitions MB91V460A

Table 3.6-1 Pin definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AL38	315	262	P00_7	D31	D31	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AJ37	314	261	P00_6	D30	D30	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AJ36	311	259	P00_5	D29	D29	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AJ35	310	258	P00_4	D28	D28	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AH36	309	257	P00_3	D27	D27	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AH35	308	256	P00_2	D26	D26	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AK38	307	255	P00_1	D25	D25	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AJ38	305	253	P00_0	D24	D24	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AH37	304	252	P01_7	D23	D23	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AG37	302	251	P01_6	D22	D22	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AG35	300	249	P01_5	D21	D21	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AG36	299	250	P01_4	D20	D20	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AF35	298	247	P01_3	D19	D19	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AF36	297	248	P01_2	D18	D18	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AH38	295	246	P01_1	D17	D17	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AF37	294	244	P01_0	D16	D16	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AG38	293	245	P02_7	D15	D15	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AE37	292	243	P02_6	D14	D14	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AE36	289	241	P02_5	D13	D13	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AD37	288	240	P02_4	D12	D12	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AD36	287	239	P02_3	D11	D11	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AC37	286	238	P02_2	D10	D10	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AF38	285	237	P02_1	D9	D9	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AE38	283	236	P02_0	D8	D8	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AD35	282	234	P03_7	D7	D7	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AC35	280	233	P03_6	D6	D6	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AB37	278	231	P03_5	D5	D5	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AC36	277	232	P03_4	D4	D4	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AA37	276	229	P03_3	D3	D3	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AB36	275	230	P03_2	D2	D2	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AD38	273	228	P03_1	D1	D1	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AC38	271	227	P03_0	D0	D0	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
AB35	272	226	X1	-	-	-	-	TO00_1	-	OSC	Stop	X1	-
AA36	270	225	X0	-	-	-	-	TO00_0	-	OSC	Stop	X0	-
AB38	267	222	X1A	-	-	-	-	TO01_1	-	OSC	Stop	X1A	-
AA35	266	221	X0A	-	-	-	-	TO01_0	-	OSC	Stop	X0A	-

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
W37	257	211	MONCLK	-	-	-	-	TC10_0	-	-	no	MONCLK	8mA
V35	256	212	P04_7	A31	A31	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
V38	255	209	P04_6	A30	A30	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
U38	253	207	P04_5	A29	A29	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
V37	252	208	P04_4	A28	A28	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
V36	251	205	P04_3	A27	A27	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
U37	250	206	P04_2	A26	A26	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
U36	249	203	P04_1	A25	A25	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
U35	248	204	P04_0	A24	A24	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
T35	246	202	P05_7	A23	A23	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
T38	243	200	P05_6	A22	A22	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
T37	242	199	P05_5	A21	A21	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
R38	241	198	P05_4	A20	A20	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
R37	240	197	P05_3	A19	A19	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
T36	239	196	P05_2	A18	A18	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
R36	237	195	P05_1	A17	A17	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
P37	236	194	P05_0	A16	A16	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
N37	234	193	P06_7	A15	A15	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
P38	233	192	P06_6	A14	A14	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
N38	231	190	P06_5	A13	A13	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
P35	230	189	P06_4	A12	A12	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
P36	229	188	P06_3	A11	A11	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
N35	228	187	P06_2	A10	A10	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
N36	227	186	P06_1	A9	A9	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
M37	226	185	P06_0	A8	A8	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
L37	224	184	P07_7	A7	A7	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
M38	221	182	P07_6	A6	A6	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
M35	220	181	P07_5	A5	A5	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
L38	219	180	P07_4	A4	A4	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
L35	218	179	P07_3	A3	A3	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
M36	217	178	P07_2	A2	A2	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
L36	215	177	P07_1	A1	A1	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
K37	214	176	P07_0	A0	A0	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
J37	212	175	P08_7	RDY	RDY	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
K38	211	174	P08_6	BRQ	BRQ	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
J38	209	172	P08_5	BGRNTX	BGRNTX	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
K35	208	171	P08_4	RDX	RDX	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
K36	207	170	P08_3	WRX3	WRX3	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
J35	206	169	P08_2	WRX2	WRX2	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
J36	205	168	P08_1	WRX1	WRX1	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
H37	202	167	P08_0	WRX0	WRX0	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
G37	200	165	P09_7	CSX7	CSX7	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
H38	199	164	P09_6	CSX6	CSX6	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
G38	197	163	P09_5	CSX5	CSX5	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
H36	196	162	P09_4	CSX4	CSX4	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
F38	195	161	P09_3	CSX3	CSX3	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
G35	194	160	P09_2	CSX2	CSX2	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
E38	193	159	P09_1	CSX1	CSX1	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
F37	192	158	P09_0	CSX0	CSX0	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
E37	190	157	P10_7	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
G36	189	156	P10_6	MCLKE	MCLKE	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
F36	187	154	P10_5	MCLKI	MCLKI	/MCLKI	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
F35	186	153	P10_4	MCLKO	MCLKO	/MCLKO	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D38	185	152	P10_3	WEX	WEX	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
E35	184	151	P10_2	BAAX	BAAX	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C38	183	150	P10_1	ASX	ASX	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D37	178	147	P10_0	SYSCLK	SYSCLK	/SYSCLK	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
E36	177	148	P11_7	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C37	176	145	P11_6	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D36	175	146	P11_5	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B36	171	143	P11_4	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C35	170	142	P11_3	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B35	169	141	P11_2	-	-	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C34	168	139	P11_1	IOWRX	IOWRX	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
A36	166	138	P11_0	IORDX	IORDX	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D34	165	137	P12_7	DEOP3	DEOP3	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
A35	164	136	P12_6	DEOTX3	DEOTX3	DEOP3	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D33	163	135	P12_5	DACKX3	DACKX3	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B34	161	134	P12_4	DREQ3	DREQ3	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C33	160	133	P12_3	DEOP2	DEOP2	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B33	159	132	P12_2	DEOTX2	DEOTX2	DEOP2	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C32	158	131	P12_1	DACKX2	DACKX2	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
A34	156	130	P12_0	DREQ2	DREQ2	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B32	155	129	P13_7	DEOP1	DEOP1	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
A33	154	128	P13_6	DEOTX1	DEOTX1	DEOP1	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B31	153	127	P13_5	DACKX1	DACKX1	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D31	149	125	P13_4	DREQ1	DREQ1	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C31	148	124	P13_3	DEOP0	DEOP0	-	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
D30	147	123	P13_2	DEOTX0	DEOTX0	DEOP0	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
C30	146	122	P13_1	DACKX0	DACKX0	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
A32	144	120	P13_0	DREQ0	DREQ0	^	-	TP04_0	U/D	CH / A / TTL	Stop	TTL (extbus)	4mA
B30	143	119	P14_7	-	ICU7/TIN7	TIN7	TTG15/7	TP00_0	U/D	CH / A	Stop	-	4mA
A31	142	118	P14_6	-	ICU6/TIN6	TIN6	TTG14/6	TP00_0	U/D	CH / A	Stop	-	4mA
B29	141	117	P14_5	-	ICU5/TIN5	TIN5	TTG13/5	TP00_0	U/D	CH / A	Stop	-	4mA

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
C29	138	116	P14_4	-	ICU4/TIN4	TIN4	TTG12/4	TP00_0	U/D	CH / A	Stop	-	4mA
D29	137	115	P14_3	-	ICU3/TIN3	TIN3	TTG11/3	TP00_0	U/D	CH / A	Stop	-	4mA
C28	136	114	P14_2	-	ICU2/TIN2	TIN2	TTG10/2	TP00_0	U/D	CH / A	Stop	-	4mA
D28	135	113	P14_1	-	ICU1/TIN1	TIN1	TTG9/1	TP00_0	U/D	CH / A	Stop	-	4mA
A30	134	112	P14_0	-	ICU0/TIN0	TIN0	TTG8/0	TP00_0	U/D	CH / A	Stop	-	4mA
A29	132	111	P15_7	-	OCU7	TOT7	-	TP00_0	U/D	CH / A	Stop	-	4mA
B28	131	110	P15_6	-	OCU6	TOT6	-	TP00_0	U/D	CH / A	Stop	-	4mA
B27	129	108	P15_5	-	OCU5	TOT5	-	TP00_0	U/D	CH / A	Stop	-	4mA
D27	127	107	P15_4	-	OCU4	TOT4	-	TP00_0	U/D	CH / A	Stop	-	4mA
C27	126	106	P15_3	-	OCU3	TOT3	-	TP00_0	U/D	CH / A	Stop	-	4mA
D26	125	105	P15_2	-	OCU2	TOT2	-	TP00_0	U/D	CH / A	Stop	-	4mA
C26	124	104	P15_1	-	OCU1	TOT1	-	TP00_0	U/D	CH / A	Stop	-	4mA
A28	122	102	P15_0	-	OCU0	TOT0	-	TP00_0	U/D	CH / A	Stop	-	4mA
B26	121	101	P16_7	-	PPG15	ATGX	-	TP00_0	U/D	CH / A	Stop	-	4mA
A27	120	100	P16_6	-	PPG14	PFM	-	TP00_0	U/D	CH / A	Stop	-	4mA
B25	119	99	P16_5	-	PPG13	SGO	-	TP00_0	U/D	CH / A	Stop	-	4mA
C25	116	97	P16_4	-	PPG12	SGA	-	TP00_0	U/D	CH / A	Stop	-	4mA
B24	115	96	P16_3	-	PPG11	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
C24	114	95	P16_2	-	PPG10	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
B23	113	94	P16_1	-	PPG9	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A26	112	93	P16_0	-	PPG8	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A25	110	92	P17_7	-	PPG7	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
D24	109	91	P17_6	-	PPG6	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
D23	107	90	P17_5	-	PPG5	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B22	105	88	P17_4	-	PPG4	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
C23	104	89	P17_3	-	PPG3	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B21	103	86	P17_2	-	PPG2	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
C22	102	87	P17_1	-	PPG1	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A24	100	85	P17_0	-	PPG0	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
D22	99	82	P18_7	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A23	98	83	P18_6	-	SCK7	ZIN3/CK7	-	TP00_0	U/D	CH / A	Stop	-	4mA
C21	97	80	P18_5	-	SOT7	BIN3	-	TP00_0	U/D	CH / A	Stop	-	4mA
A22	94	79	P18_4	-	SIN7	AIN3	-	TP00_0	U/D	CH / A	Stop	-	4mA
D21	93	77	P18_3	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A21	92	78	P18_2	-	SCK6	ZIN2/CK6	-	TP00_0	U/D	CH / A	Stop	-	4mA
D20	91	75	P18_1	-	SOT6	BIN2	-	TP00_0	U/D	CH / A	Stop	-	4mA
B20	90	76	P18_0	-	SIN6	AIN2	-	TP00_0	U/D	CH / A	Stop	-	4mA
C20	89	73	P19_7	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A20	88	74	P19_6	-	SCK5	CK5	-	TP00_0	U/D	CH / A	Stop	-	4mA
C19	87	71	P19_5	-	SOT5	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A19	86	72	P19_4	-	SIN5	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
D19	85	70	P19_3	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
B19	84	69	P19_2	-	SCK4	CK4	-	TP00_0	U/D	CH / A	Stop	-	4mA
D18	83	68	P19_1	-	SOT4	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A18	82	67	P19_0	-	SIN4	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A17	80	65	P20_7	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B18	79	64	P20_6	-	SCK3	ZIN1/CK3	-	TP00_0	U/D	CH / A	Stop	-	4mA
C18	78	63	P20_5	-	SOT3	BIN1	-	TP00_0	U/D	CH / A	Stop	-	4mA
B17	77	62	P20_4	-	SIN3	AIN1	-	TP00_0	U/D	CH / A	Stop	-	4mA
C17	76	61	P20_3	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
D17	75	60	P20_2	-	SCK2	ZIN0/CK2	-	TP00_0	U/D	CH / A	Stop	-	4mA
D16	73	58	P20_1	-	SOT2	BIN0	-	TP00_0	U/D	CH / A	Stop	-	4mA
A16	70	57	P20_0	-	SIN2	AIN0	-	TP00_0	U/D	CH / A	Stop	-	4mA
B16	69	55	P21_7	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A15	68	56	P21_6	-	SCK1	CK1	-	TP00_0	U/D	CH / A	Stop	-	4mA
B15	67	53	P21_5	-	SOT1	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
C16	66	54	P21_4	-	SIN1	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
C15	64	52	P21_3	-	-	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B14	63	51	P21_2	-	SCK0	CK0	-	TP00_0	U/D	CH / A	Stop	-	4mA
B13	61	49	P21_1	-	SOT0	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A14	60	50	P21_0	-	SIN0	^	-	TP00_0	U/D	CH / A	Stop	-	4mA
A13	58	47	P22_7	-	SCL1	-	-	TP02_0	-	CH / A	Stop	I2C	3mA
D14	57	46	P22_6	-	SDA1	-	INT15	TP02_0	-	CH / A	Stop	I2C	3mA
C14	56	45	P22_5	-	SCL0	-	-	TP02_0	-	CH / A	Stop	I2C	3mA
D13	55	44	P22_4	-	SDA0	-	INT14	TP02_0	-	CH / A	Stop	I2C	3mA
C13	54	42	P22_3	-	TX5	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B12	53	43	P22_2	-	RX5	-	INT13	TP00_0	U/D	CH / A	Stop	-	4mA
B11	51	41	P22_1	-	TX4	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A12	48	39	P22_0	-	RX4	-	INT12	TP00_0	U/D	CH / A	Stop	-	4mA
D12	47	38	P23_7	-	TX3	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A11	46	37	P23_6	-	RX3	-	INT11	TP00_0	U/D	CH / A	Stop	-	4mA
D11	45	36	P23_5	-	TX2	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
C12	44	35	P23_4	-	RX2	-	INT10	TP00_0	U/D	CH / A	Stop	-	4mA
C11	42	34	P23_3	-	TX1	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B10	41	33	P23_2	-	RX1	-	INT9	TP00_0	U/D	CH / A	Stop	-	4mA
B9	39	32	P23_1	-	TX0	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
A10	38	31	P23_0	-	RX0	-	INT8	TP00_0	U/D	CH / A	Stop	-	4mA
A9	36	29	P24_7	-	INT7	-	SCL3	TP02_0	-	CH / A	Stop	I2C	3mA
D10	35	28	P24_6	-	INT6	-	SDA3	TP02_0	-	CH / A	Stop	I2C	3mA
C10	34	26	P24_5	-	INT5	-	SCL2	TP02_0	-	CH / A	Stop	I2C	3mA
D9	33	27	P24_4	-	INT4	-	SDA2	TP02_0	-	CH / A	Stop	I2C	3mA
C9	32	25	P24_3	-	INT3	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B8	29	24	P24_2	-	INT2	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
B7	27	23	P24_1	-	INT1	-	-	TP00_0	U/D	CH / A	Stop	-	4mA

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
A8	26	20	P24_0	-	INT0	-	-	TP00_0	U/D	CH / A	Stop	-	4mA
D2	688	576	P25_7	-	SMC2M5	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
E3	687	575	P25_6	-	SMC2P5	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
C1	685	573	P25_5	-	SMC1M5	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
E4	684	572	P25_4	-	SMC1P5	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
D1	683	571	P25_3	-	SMC2M4	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
F4	682	570	P25_2	-	SMC2P4	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
E2	680	568	P25_1	-	SMC1M4	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
F2	678	567	P25_0	-	SMC1P4	-	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
F3	679	566	P26_7	-	SMC2M3	AN31	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
G3	677	565	P26_6	-	SMC2P3	AN30	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
E1	675	562	P26_5	-	SMC1M3	AN29	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
G2	674	563	P26_4	-	SMC1P3	AN28	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
F1	673	560	P26_3	-	SMC2M2	AN27	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
H2	672	561	P26_2	-	SMC2P2	AN26	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
H4	668	558	P26_1	-	SMC1M2	AN25	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
J4	666	557	P26_0	-	SMC1P2	AN24	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
H3	667	556	P27_7	-	SMC2M1	AN23	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
J3	665	555	P27_6	-	SMC2P1	AN22	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
G1	663	552	P27_5	-	SMC1M1	AN21	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
J2	662	553	P27_4	-	SMC1P1	AN20	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
H1	661	550	P27_3	-	SMC2M0	AN19	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
K2	660	551	P27_2	-	SMC2P0	AN18	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
K3	657	547	P27_1	-	SMC1M0	AN17	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
K4	656	548	P27_0	-	SMC1P0	AN16	-	TP05_0	-	CH / A	Stop	SMC / AN	30mA
L3	655	545	ALARM_1	-	-	-	-	TA02_0	-	AN IN	-	Ana In	-
L4	654	546	ALARM_0	-	-	-	-	TA02_0	-	AN IN	-	Ana In	-
A5	20	14	P28_7	-	AN15	-	DA1	TP01_0	-	CH / A	Stop	AN / DA	4mA
B5	17	13	P28_6	-	AN14	-	DA0	TP01_0	-	CH / A	Stop	AN / DA	4mA
C7	16	12	P28_5	-	AN13	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
C6	14	10	P28_4	-	AN12	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
D6	13	11	P28_3	-	AN11	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
A4	12	8	P28_2	-	AN10	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
D5	11	9	P28_1	-	AN9	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
A3	10	6	P28_0	-	AN8	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
B4	5	4	P29_7	-	AN7	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
C5	4	3	P29_6	-	AN6	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
B3	3	2	P29_5	-	AN5	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
C4	2	1	P29_4	-	AN4	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
C2	690	578	P29_3	-	AN3	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
D3	689	577	P29_2	-	AN2	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
J1	653	544	P29_1	-	AN1	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
L2	650	543	P29_0	-	AN0	-	-	TP03_0	U/D	CH / A	Stop	AN	4mA
K1	651	542	P30_7	-	V3	-	-	TP08_0	-	CH / A	Stop	LCD V3	4mA
M2	648	541	P30_6	-	V2	-	-	TP07_0	-	CH / A	Stop	LCD V0,V1,V2	4mA
M4	646	540	P30_5	-	V1	-	-	TP07_0	-	CH / A	Stop	LCD V0,V1,V2	4mA
M3	645	538	P30_4	-	V0	-	-	TP07_0	-	CH / A	Stop	LCD V0,V1,V2	4mA
N4	644	539	P30_3	-	COM3	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
N3	643	536	P30_2	-	COM2	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
L1	641	534	P30_1	-	COM1	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
N2	640	535	P30_0	-	COM0	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
M1	639	532	P31_7	-	SEG39	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
P2	638	533	P31_6	-	SEG38	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
P3	635	529	P31_5	-	SEG37	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
R2	634	530	P31_4	-	SEG36	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
R3	633	527	P31_3	-	SEG35	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
T2	632	528	P31_2	-	SEG34	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
N1	631	526	P31_1	-	SEG33	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
P1	629	524	P31_0	-	SEG32	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
R4	628	525	P32_7	-	SEG31	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
T4	626	523	P32_6	-	SEG30	SCK15	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
U2	624	522	P32_5	-	SEG29	SOT15	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
T3	623	521	P32_4	-	SEG28	SIN15	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
V2	622	520	P32_3	-	SEG27	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
U3	621	519	P32_2	-	SEG26	SCK14	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
R1	619	517	P32_1	-	SEG25	SOT14	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
U4	618	516	P32_0	-	SEG24	SIN14	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
T1	617	515	P33_7	-	SEG23	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
V3	616	514	P33_6	-	SEG22	SCK13	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
U1	613	512	P33_5	-	SEG21	SOT13	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
V4	612	511	P33_4	-	SEG20	SIN13	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
V1	611	510	P33_3	-	SEG19	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
W4	610	509	P33_2	-	SEG18	SCK12	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
W2	609	508	P33_1	-	SEG17	SOT12	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
W3	608	507	P33_0	-	SEG16	SIN12	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
W1	607	506	P34_7	-	SEG15	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
Y3	606	505	P34_6	-	SEG14	SCK11	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
Y1	605	503	P34_5	-	SEG13	SOT11	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
Y4	604	504	P34_4	-	SEG12	SIN11	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
Y2	603	501	P34_3	-	SEG11	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AA4	602	502	P34_2	-	SEG10	SCK10	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AA1	601	499	P34_1	-	SEG9	SOT10	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AB1	599	497	P34_0	-	SEG8	SIN10	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AA2	598	498	P35_7	-	SEG7	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AA3	597	495	P35_6	-	SEG6	SCK9	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AB2	596	496	P35_5	-	SEG5	SOT9	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AB3	595	493	P35_4	-	SEG4	SIN9	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AB4	594	494	P35_3	-	SEG3	-	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AC4	592	492	P35_2	-	SEG2	SCK8	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AC1	589	489	P35_1	-	SEG1	SOT8	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AC2	588	490	P35_0	-	SEG0	SIN8	-	TP06_0	-	CH / A	Stop	LCD COM/SEG	4mA
AD1	587	487	INITX	-	-	-	-	TC02_0	Up	CH (old)	no	MCU control	-
AD2	586	488	RSTX	-	-	-	-	TC01_0	Up	CH (old)	no	MCU control	-
AC3	585	486	HSTX	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AD3	583	484	NMIX	-	-	-	-	TC01_0	Up	CH (old)	no	MCU control	-
AE2	582	485	MD_2	-	-	-	-	TC02_0	-	CH (old)	no	MCU control	-
AF2	580	483	MD_1	-	-	-	-	TC02_0	-	CH (old)	no	MCU control	-
AE1	579	481	MD_0	-	-	-	-	TC02_0	-	CH (old)	no	MCU control	-
AF1	577	479	ADC_SEL	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AE4	576	480	ALARM_SEL	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AE3	575	477	SRAM_SFX	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AF4	574	478	FSC_DISABLE	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AF3	573	475	CSV_KILL	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AG2	572	476	HWWDG_KILL	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AH2	570	474	FIX_ENX	-	-	-	-	TC00_0	-	CH (old)	no	MCU control	-
AG4	566	472	EDSU_BREAKX	-	-	-	-	TC01_0	Up	CH (old)	no	MCU control	-
AG1	567	471	ICLK	-	-	-	-	TE22_0	-	C	no	Tool	8mA
AH1	565	469	ICD_3	-	-	-	-	TE21_0	-	C	no	Tool	4mA
AH4	564	470	ICD_2	-	-	-	-	TE21_0	-	C	no	Tool	4mA
AG3	563	467	ICD_1	-	-	-	-	TE21_0	-	C	no	Tool	4mA
AH3	561	465	ICD_0	-	-	-	-	TE21_0	-	C	no	Tool	4mA
AJ2	560	466	ICS_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AK2	558	464	ICS_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AJ1	557	462	ICS_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AK1	555	460	BREAK	-	-	-	-	TE01_0	-	C	no	Tool	-
AJ4	554	461	PLEVEL	-	-	-	-	TE02_0	-	C	no	Tool	-
AJ3	553	458	EXRAM	-	-	-	-	TE00_0	-	C	no	Tool	-
AK4	552	459	TRSTX	-	-	-	-	TE03_0	-	C	no	Tool	-
AK3	551	457	TCLK	-	-	-	-	TE11_0	-	-	no	Tool	8mA
AL2	548	456	TOEX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM2	546	454	TWRX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AL1	545	453	TCE1X	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM1	543	450	TADSCX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AL3	542	451	TAD_15	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AN1	541	448	TAD_14	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM4	540	449	TAD_13	-	-	-	-	TE10_0	-	-	no	Tool	4mA

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AP1	539	446	TAD_12	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AN2	538	447	TAD_11	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AP2	536	445	TAD_10	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM3	535	444	TAD_9	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AN3	533	443	TAD_8	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AN4	532	442	TAD_7	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR1	531	441	TAD_6	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AP4	530	440	TAD_5	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT1	529	439	TAD_4	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR2	524	436	TAD_3	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AP3	523	435	TAD_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT2	522	434	TAD_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR3	521	433	TAD_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU3	517	432	TDT_68	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT4	516	430	TDT_67	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU4	515	431	TDT_66	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT5	514	428	TDT_65	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV3	512	426	TDT_64	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR5	511	427	TDT_63	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV4	510	424	TDT_62	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR6	509	425	TDT_61	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU5	507	423	TDT_60	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT6	506	421	TDT_59	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU6	505	422	TDT_58	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT7	504	419	TDT_57	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV5	502	417	TDT_56	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU7	501	420	TDT_55	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV6	500	416	TDT_54	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU8	499	418	TDT_53	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR8	495	415	TDT_52	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT8	494	412	TDT_51	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR9	493	413	TDT_50	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT9	492	410	TDT_49	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV7	490	408	TDT_48	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU9	489	409	TDT_47	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV8	488	406	TDT_46	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU10	487	407	TDT_45	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT10	484	404	TDT_44	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR10	483	405	TDT_43	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT11	482	402	TDT_42	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR11	481	403	TDT_41	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV9	480	401	TDT_40	-	-	-	-	TE20_0	-	C	no	Tool	4mA

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AV10	478	400	TDT_39	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU11	477	399	TDT_38	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU12	475	398	TDT_37	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR12	473	396	TDT_36	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT12	472	395	TDT_35	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR13	471	394	TDT_34	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT13	470	392	TDT_33	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV11	468	390	TDT_32	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU13	467	391	TDT_31	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV12	466	388	TDT_30	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU14	465	389	TDT_29	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT14	462	386	TDT_28	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU15	461	385	TDT_27	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT15	460	383	TDT_26	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU16	459	384	TDT_25	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV13	458	382	TDT_24	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV14	456	381	TDT_23	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR15	455	380	TDT_22	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR16	453	379	TDT_21	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU17	451	377	TDT_20	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT16	450	376	TDT_19	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU18	449	375	TDT_18	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT17	448	374	TDT_17	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV15	446	371	TDT_16	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR17	445	372	TDT_15	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV16	444	369	TDT_14	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT18	443	370	TDT_13	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV17	440	368	TDT_12	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR18	439	367	TDT_11	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV18	438	366	TDT_10	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR19	437	365	TDT_9	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU19	436	364	TDT_8	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT19	435	363	TDT_7	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV19	434	362	TDT_6	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT20	433	361	TDT_5	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV20	432	360	TDT_4	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR20	431	359	TDT_3	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU20	430	358	TDT_2	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR21	429	357	TDT_1	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV21	428	356	TDT_0	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV22	426	354	EMRAM	-	-	-	-	TE00_0	-	C	no	Tool	-
AU21	425	353	ECSX	-	-	-	-	TE10_0	-	-	no	Tool	4mA

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AT21	424	351	EWRX_3	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU22	423	352	EWRX_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT22	422	349	EWRX_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR22	421	350	EWRX_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR23	419	348	EECSX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV23	416	346	EEWEX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU23	415	345	EEOEX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV24	414	344	EEBEX_3	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU24	413	343	EEBEX_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT23	412	342	EEBEX_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT24	410	340	EEBEX_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU25	409	339	EEA_20	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU26	407	338	EEA_19	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV25	406	336	EEA_18	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV26	404	335	EEA_17	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR25	403	334	EEA_16	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT25	402	333	EEA_15	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR26	401	332	EEA_14	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT26	400	330	EEA_13	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU27	399	331	EEA_12	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU28	397	329	EEA_11	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV27	394	328	EEA_10	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR27	393	327	EEA_9	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV28	392	326	EEA_8	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR28	391	325	EEA_7	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT27	390	324	EEA_6	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT28	388	322	EEA_5	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU29	387	321	EEA_4	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AU30	385	320	EEA_3	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV29	384	319	EEA_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AV30	382	317	EEA_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AR29	381	316	EEA_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AT29	380	315	EED_31	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR30	379	314	EED_30	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT30	378	313	EED_29	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU31	375	311	EED_28	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU32	373	310	EED_27	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV31	372	309	EED_26	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV32	370	307	EED_25	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT31	369	306	EED_24	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV33	368	305	EED_23	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR32	367	304	EED_22	-	-	-	-	TE20_0	-	C	no	Tool	4mA

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AV34	366	303	EED_21	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU33	365	302	EED_20	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU34	363	300	EED_19	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT32	362	301	EED_18	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT33	360	299	EED_17	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR33	359	297	EED_16	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV35	358	298	EED_15	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR34	357	295	EED_14	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AV36	356	296	EED_13	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU35	351	292	EED_12	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT34	350	291	EED_11	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AU36	349	290	EED_10	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT35	348	289	EED_9	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT37	344	288	EED_8	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR36	343	287	EED_7	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR37	342	286	EED_6	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AP36	341	285	EED_5	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AT38	339	283	EED_4	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AP35	338	282	EED_3	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AR38	337	281	EED_2	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AN35	336	280	EED_1	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AP37	334	279	EED_0	-	-	-	-	TE20_0	-	C	no	Tool	4mA
AN36	333	278	IHIT_3	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AN37	332	277	IHIT_2	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM36	331	276	IHIT_1	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AP38	329	275	IHIT_0	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM37	328	274	FLASH_RDY	-	-	-	-	TE00_0	-	C	no	Tool	-
AN38	327	273	FLASH_ALE	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AL37	326	272	FLASH_SYNC	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AL35	322	269	FLASH_ATDIN	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AL36	321	268	FLASH_EQIN	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AK35	320	267	FLASH_RD32	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AK36	319	266	FLASH_BYTEX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AM38	317	265	FLASH_SECUR	-	-	-	-	TE10_0	-	-	no	Tool	4mA
AK37	316	263	FLASH_FRSTX	-	-	-	-	TE10_0	-	-	no	Tool	4mA
0.0	637	531	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
0.0	615	513	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
0.0	591	491	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
0.0	569	473	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AR7	313	260	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AP11	291	242	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AP19	269	224	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AP23	245	201	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AP31	223	183	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AM35	201	166	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AH34	179	149	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
AD34	172	144	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
T34	152	126	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
H34	130	109	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
H35	118	98	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
E32	96	81	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
E28	72	59	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
E20	50	40	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
E12	28	22	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
D8	6	5	VDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
M5	642	537	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
T5	620	518	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
Y5	600	500	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
AD5	578	482	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
AJ34	306	254	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
AE34	284	235	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
AA34	268	223	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
W34	254	210	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
R34	232	191	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
L34	210	173	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
G34	188	155	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
B37	167	140	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E31	145	121	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E27	123	103	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E23	101	84	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E19	81	66	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E15	59	48	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
E11	37	30	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
C3	9	7	VSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
0.0	681	569	HVDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
0.0	671	559	HVDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
0.0	659	549	HVDD5	-	-	-	-	TS02_0	-	-	-	VDD 5V	-
B2	686	574	HVSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
F5	676	564	HVSS5	-	-	-	-	TS00_0	-	-	-	VSS	-
H5	664	554	HVSS5	-	-	-	-	TS00_0	-	-	-	VSS	-

MB91460 Series

3.6.Pin Definitions MB91V460A

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
E9	25	21	AVSS	-	-	-	-	TA03_0	-	AVSS	-	AVSS	-
C8	23	19	AVRL	-	-	-	-	TA01_0	-	AVRH/L	-	AVRH/L	-
A7	24	18	AVRH5	-	-	-	-	TA01_0	-	AVRH/L	-	AVRH/L	-
D7	21	17	AVCC5	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
A6	22	16	AVRH3	-	-	-	-	TA01_0	-	AVRH/L	-	AVRH/L	-
B6	19	15	AVCC3	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
Y38	261	216	VDD5R	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
W36	260	215	VDD5R	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
W38	259	214	VDD5R	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
W35	258	213	VDD5R	-	-	-	-	TA00_0	-	AVCC	-	AVCC	-
AA38	265	220	VCC3C	-	-	-	-	TA10_0	-	ANA OUT	-	VCC3C	-
Y35	264	219	VCC3C	-	-	-	-	TA10_0	-	ANA OUT	-	VCC3C	-
Y37	263	218	VCC3C	-	-	-	-	TA10_0	-	ANA OUT	-	VCC3C	-
Y36	262	217	VCC3C	-	-	-	-	TA10_0	-	ANA OUT	-	VCC3C	-
0.0	562	468	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	549	455	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	527	437	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	497	414	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	474	397	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	463	387	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
0.0	452	378	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
J5	420	347	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
N5	411	341	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
U5	389	323	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
AD4	376	312	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
AE5	354	294	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
AN5	324	271	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
AP9	323	270	VDD3	-	-	-	-	TS01_0	-	-	-	VDD	-
AH5	556	463	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AK5	544	452	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AT3	528	438	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AU2	513	429	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP8	491	411	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP12	469	393	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP16	447	373	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP20	427	355	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP24	405	337	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-

JEDEC	Pin	Pad	I/O	Function	PFR=1	EPFR=1	Special	Type	Pull Up/Down	CMOS/ CMOS Hyst/ Auto/ TTL	Input Stop	Usage	Output
AP28	383	318	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AP30	371	308	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AT36	355	293	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AU37	340	284	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-
AL34	318	264	VSS3	-	-	-	-	TS00_0	-	-	-	VSS	-

MB91460 Series**3.7. I/O Circuit Types MB91V460A**

The table below describes the circuit types which are used on the evaluation device MB91V460A

Please refer to the datasheets for information about the circuit type of each pin used on the flash devices.

Table 3.7-1 I/O circuit types MB91V460A

Type	Pull Up/ Pull Down	CMOS/ CMOS Hyst./ Automotive/ TTL	Input Stop	Usage	Output drive
TP00_0	Up / Dn	CH / A	yes	-	4 mA
TP01_0	-	CH / A	yes	A/D in / D/A out	4 mA
TP02_0	-	CH / A	yes	I2C	3 mA
TP03_0	Up / Dn	CH / A	yes	A/D in	4 mA
TP04_0	Up / Dn	CH / A / TTL	yes	External Bus	4 mA
TP05_0	-	CH / A	yes	SMC / A/D in	30 mA
TP06_0	-	CH / A	yes	LCD COM/SEG	4 mA
TP07_0	-	CH / A	yes	LCD V0/V1/V2	4 mA
TP08_0	-	CH / A	yes	LCD V3	4 mA
TC00_0	-	CH	no	MCU control	-
TC01_0	Up	CH	no	MCU control	-
TC02_0	Up	CH	no	MCU control	-
TC02_1	-	CH	no	MD0/MD1/MD2	-
TC10_0	-	-	no	MONCLK	8 mA
TA00_0	-	-	-	AVCC	-
TA01_0	-	-	-	AVRH/AVRL	-
TA02_0	-	-	-	A/D in	-
TA03_0	-	-	-	AVSS	-
TA10_0	-	-	-	VDD 3V	-
TO00_0	-	-	yes	X0	-
TO00_1	-	-	yes	X1	-
TO01_0	-	-	yes	X0A	-
TO01_1	-	-	yes	X1A	-
TE00_0	-	C	no	Tool	-
TE01_0	Dn (ctrl)	C	no	Tool	-
TE02_0	Up	C	no	Tool	-

TE03_0	Up	C	no	Tool	-
TE10_0	-	-	no	Tool	4 mA
TE11_0	-	-	no	Tool	8 mA
TE20_0	-	C	no	Tool	4 mA
TE21_0	Dn (ctrl)	C	no	Tool	4 mA
TE22_0	-	C	no	Tool	8 mA
TS00_0	-	-	-	VSS	
TS01_0	-	-	-	VDD	
TS02_0	-	-	-	VDD 5V	

MB91460 Series

3.8. Pin Definitions MB91FV460B

3.8.1 MB91460 Series Modes

MB91FV460B supports the pin multiplexing and the interrupt relocation of all MB91460 series devices. The pins SER_MD_2, SER_MD_1 and SER_MD_0 ("Series Mode") are used to setup the emulation of the different devices.

Table 3.8-1 MB91FV460B Series Modes

SER_MD			Mode name	Emulated Devices	Series Mode
2	1	0			
0	0	0	default	MB91F464Ax MB91F465Kx MB91F465Cx MB91F465Dx MB91F467Cx MB91F467Dx MB91F469Gx	Default mode for emulation of old devices. External bus is enabled after INIT (PFR,EPFR=1)
1	1	1	new		Default mode for emulation of new devices. External bus is disabled after INIT (PFR,EPFR=0) in internal vector fetch mode (MD=000)
0	0	1	P/T	MB91F465Px MB91F467Px MB91F467Tx MB91F469Tx	Pin multiplexing for MB91460P and T series (see data sheets), includes that external bus is disabled after INIT in internal vector fetch mode (MD=000).
0	1	0	461	MB91461	MB91461 mode: - supports special clock features of MB91461 - the I2C pins always run as open drain pins, in GPIO mode too. However, MB91FV460B does not support the separated power domains of MB91461 IO ring.
0	1	1	B	MB91F464BB MB91F465BB MB91F466BA MB91F467BA	Pin multiplexing for MB91460B series, emulates the "no external bus" mode (MD_3=0). For MD3=1 (with external bus) use series mode "default".
1	0	0	M	MB91F467M	Interrupt relocation for MB91460M series
1	0	1	Q	MB91F469Qx	Interrupt relocation for MB91460Q series, external bus is disabled after INIT in internal vector fetch mode (MD=000).
1	1	0	-		-

Note : The PPMUX and PPMUX2 registers, used in P mode for I/O relocation, can be accessed in any series mode. There is no access restriction. The user has to take care that these registers are not written in other series modes than P.

Note : The pins SER_MD_2 to SER_MD_0 have internal pull-down.

3.8.2 General Purpose Port and Resource Function IO Names

The following table explains the naming of ports and resource terminals, which are multiplexed on general purpose IO pins. For the description of dedicated IO pins, please refer to section [“Dedicated and Special Pins” on page 165](#).

Table 3.8-2 General purpose port and resource function IO names

Pin name	Block	Function
GPxx_n	GPIO	General purpose IO port
Dn	External Bus	External Bus Data line
An	External Bus	External Bus Address line
WRXn	External Bus	External Bus Write strobe output
RDX	External Bus	External Bus Read strobe output
BGRNTX	External Bus	External Bus Release reception output
BRQ	External Bus	External Bus Release request input
RDY	External Bus	External Bus Ready input
CSXn	External Bus	External Bus Chip select output
ASX	External Bus	External Bus Address strobe output
BAAX	External Bus	External Bus Burst address advance output
WEX	External Bus	External Bus Write enable output
MCLKO	External Bus	External Bus Clock output for memory
MCLKI	External Bus	External Bus Clock input for memory
MCLKE	External Bus	External Bus Clock enable signal for memory
SYSCLK	External Bus	External Bus System clock output
IOWRX	DMA	Output for DMA memory to I/O fly-by transfer.
IORDX	DMA	Output for DMA I/O to memory fly-by transfer.
DREQn	DMA	Transfer request input
DACKXn	DMA	Transfer request acknowledgement output
DEOPn	DMA	End of transfer output
DEOTXn	DMA	Transfer stop request input
INTn	External Interrupt	Interrupt input
INTn-M, INTn-Q	External Interrupt	Relocated interrupt input (in series mode M and Q)
ICUn	Input Capture	Input Capture input pin of ICU
TTGn	Programmable Pulse Generator	External trigger input pins of PPG timer
OCUn	Output Compare	Output compare output
TINn	Reload Timer	External trigger input of Reload Timer
TOTn	Reload Timer	Reload timer output
PFM	Pulse Frequency Modulator	Pulse frequency modulator output
SGO	Sound Generator	Sound Generator output
SGA	Sound Generator	Sound Generator output

MB91460 Series

Pin name	Block	Function
AINn	Up-/Down Counter	Up-/Down Counter input A
BINn	Up-/Down Counter	Up-/Down Counter input B
ZINn	Up-/Down Counter	Up-/Down Counter input Z
SCKn	LIN-USART	LIN-USART serial clock input/output
SINn	LIN-USART	LIN-USART data input
SOTn	LIN-USART	LIN-USART data output
SCLn	I ² C	I ² C bus clock input/output
SDAn	I ² C	I ² C bus data input/output
CKn	Free-Run Timer	External clock input of free-run timer
SMC1Pn, SMC2Pn	Stepper Motor Control	Stepper Motor Controller plus outputs
SMC1Mn, SMC2Mn	Stepper Motor Control	Stepper Motor Controller minus outputs
ALARM_n	ALARM Comparator	ALARM Comparator analog input
ATGX	A/D Converter	A/D converter external trigger input
ANn	A/D Converter	A/D Converter analog input
DAn	D/A Converter	D/A Converter analog output
MLBCLK	Media LB	Media LB clock output
MLBSIG	Media LB	Media LB data input/output
MLBDAT	Media LB	Media LB data input/output
ISCK0	I ² S	I ² S clock input/output
WS0	I ² S	I ² S WS output
SDn	I ² S	I ² S data input/output
RXDA, RXDB	FlexRay	FlexRay channel A/B receiver inputs
TXDA, TXDB	FlexRay	FlexRay channel A/B transmitter outputs
TXENA, TXENB	FlexRay	FlexRay channel A/B transmit enable outputs
RCKn	APIX	APIX Sideband receiver clocks
RDAn0	APIX	APIX Sideband receiver data 0
RDAn1	APIX	APIX Sideband receiver data 1
TCLKIn	APIX	APIX Sideband transmitter clock input
TDAn0	APIX	APIX Sideband transmitter data 0
TDAn1	APIX	APIX Sideband transmitter data 1
TCLKOn	APIX	APIX Sideband transmitter clock output
STOPWT	FlexRay	FlexRay STOPWATCH input

3.8.3 User Ports in Default Series Mode

The following table shows the General Purpose IO ports and their assigned resource functions in default series mode. Make sure that the PPMUX and PPMUX2 registers are not set when the default mode is used.

Table 3.8-3 User ports in default series mode

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP00_0	AP35	622	I/O	A	VDD35	GP00_0	D24		
GP00_1	AR36	498				GP00_1	D25		
GP00_2	AT37	366				GP00_2	D26		
GP00_3	AU38	226				GP00_3	D27		
GP00_4	AL35	625				GP00_4	D28		
GP00_5	AK34	741				GP00_5	D29		
GP00_6	AR38	228				GP00_6	D30		
GP00_7	AN37	369				GP00_7	D31		
GP01_0	AL36	502	I/O	A	VDD35	GP01_0	D16		
GP01_1	AM37	370				GP01_1	D17		
GP01_2	AP39	82				GP01_2	D18		
GP01_3	AL37	371				GP01_3	D19		
GP01_4	AK37	372				GP01_4	D20		
GP01_5	AL38	232				GP01_5	D21		
GP01_6	AG36	506				GP01_6	D22		
GP01_7	AF35	630				GP01_7	D23		
GP02_0	AP36	499	I/O	A	VDD35	GP02_0	D8		
GP02_1	AM35	624				GP02_1	D9		
GP02_2	AN36	500				GP02_2	D10		
GP02_3	AP37	368				GP02_3	D11		
GP02_4	AT39	80				GP02_4	D12		
GP02_5	AP38	229				GP02_5	D13		
GP02_6	AR39	81				GP02_6	D14		
GP02_7	AJ35	627				GP02_7	D15		
GP03_0	AH36	505	I/O	A	VDD35	GP03_0	D0		
GP03_1	AM38	231				GP03_1	D1		
GP03_2	AG35	629				GP03_2	D2		
GP03_3	AM39	84				GP03_3	D3		
GP03_4	AL39	85				GP03_4	D4		
GP03_5	AH37	374				GP03_5	D5		
GP03_6	AF36	507				GP03_6	D6		
GP03_7	AE36	508				GP03_7	D7		

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP04_0	K39	106	I/O	A	VDD35	GP04_0	A24		
GP04_1	M37	390				GP04_1	A25		
GP04_2	P36	519				GP04_2	A26		
GP04_3	P34	757				GP04_3	A27		
GP04_4	K38	253				GP04_4	A28		
GP04_5	K37	392				GP04_5	A29		
GP04_6	M36	521				GP04_6	A30		
GP04_7	G38	256				GP04_7	A31		
GP05_0	K36	523	I/O	A	VDD35	GP05_0	A16		
GP05_1	D39	112				GP05_1	A17		
GP05_2	E38	258				GP05_2	A18		
GP05_3	J35	647				GP05_3	A19		
GP05_4	H35	648				GP05_4	A20		
GP05_5	D38	259				GP05_5	A21		
GP05_6	E37	397				GP05_6	A22		
GP05_7	E36	528				GP05_7	A23		
GP06_0	T35	640	I/O	B1	VDD35	GP06_0	A8		
GP06_1	R35	641				GP06_1	A9		
GP06_2	L38	252				GP06_2	A10		
GP06_3	N36	520				GP06_3	A11		
GP06_4	P35	642				GP06_4	A12		
GP06_5	J39	107				GP06_5	A13		
GP06_6	N35	643				GP06_6	A14		
GP06_7	G39	109				GP06_7	A15		
GP07_0	M35	644	I/O	B1	VDD35	GP07_0	A0		
GP07_1	J37	393				GP07_1	A1		
GP07_2	L35	645				GP07_2	A2		
GP07_3	K35	646				GP07_3	A3		
GP07_4	F37	396				GP07_4	A4		
GP07_5	K34	761				GP07_5	A5		
GP07_6	G36	526				GP07_6	A6		
GP07_7	C38	260				GP07_7	A7		

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP08_0	AG37	375	I/O	A	VDD35	GP08_0	WRX0		
GP08_1	AH38	235				GP08_1	WRX1		
GP08_2	AG38	236				GP08_2	WRX2		
GP08_3	AD35	632				GP08_3	WRX3		
GP08_4	AD37	378				GP08_4	RDX		
GP08_5	AF39	90				GP08_5	BGRNTX		
GP08_6	AC35	633				GP08_6	BRQ		
GP08_7	AE39	91				GP08_7	RDY		
GP09_0	AJ39	87	I/O	A	VDD35	GP09_0	CSX0		
GP09_1	AH39	88				GP09_1	CSX1		
GP09_2	AD36	509				GP09_2	CSX2		
GP09_3	AG39	89				GP09_3	CSX3		
GP09_4	AE38	238				GP09_4	CSX4		
GP09_5	AD38	239				GP09_5	CSX5		
GP09_6	AA38	242				GP09_6	CSX6		
GP09_7	AA36	512				GP09_7	CSX7		
GP10_0	AB35	634	I/O	A	VDD35	GP10_0	SYSCLK	/ SYSCLK	
GP10_1	AD39	92				GP10_1	ASX		
GP10_2	AC38	240				GP10_2	BAAX		
GP10_3	AB38	241				GP10_3	WEX		
GP10_4	Y36	513				GP10_4	MCLKO	/ MCLKO	
GP10_5	Y34	751				GP10_5	MCLKI	/ MCLKI	
GP10_6	W38	244				GP10_6	MCLKE		
GP10_7	U39	99				GP10_7			
GP11_0	AC39	93	I/O	A	VDD35	GP11_0	IORDX		
GP11_1	Y38	243				GP11_1	IOWRX		
GP11_2	Y37	382				GP11_2			
GP11_3	Y39	96				GP11_3			
GP11_4	Y35	636				GP11_4			
GP11_5	V39	98				GP11_5			
GP11_6	V38	245				GP11_6			
GP11_7	U38	246				GP11_7			

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP12_0	W36	514	I/O	A	VDD35	GP12_0	DREQ2		
GP12_1	T39	100				GP12_1	DACKX2		
GP12_2	T38	247				GP12_2	DEOTX2	DEOP2	
GP12_3	R38	248				GP12_3	DEOP2		
GP12_4	T37	386				GP12_4	DREQ3		
GP12_5	N38	250				GP12_5	DACKX3		
GP12_6	T36	517				GP12_6	DEOTX3	DEOP3	
GP12_7	L39	105				GP12_7	DEOP3		
GP13_0	W35	637	I/O	A	VDD35	GP13_0	DREQ0		
GP13_1	V34	753				GP13_1	DACKX0		
GP13_2	V35	638				GP13_2	DEOTX0	DEOP0	
GP13_3	U37	385				GP13_3	DEOP0		
GP13_4	U36	516				GP13_4	DREQ1		
GP13_5	M39	104				GP13_5	DACKX1		
GP13_6	R36	518				GP13_6	DEOTX1	DEOP1	
GP13_7	T34	755				GP13_7	DEOP1		

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP14_0	D36	529	I/O	A	VDD5	GP14_0	ICU0 TIN0/8 TTG24/16/8/0		
GP14_1	C37	399				GP14_1	ICU1 TIN1/9 TTG25/17/9/1		
GP14_2	C36	400				GP14_2	ICU2 TIN2/10 TTG26/18/10/2		
GP14_3	E33	653				GP14_3	ICU3 TIN3/11 TTG27/19/11/3		
GP14_4	C35	401				GP14_4	ICU4 TIN4/12 TTG28/20/12/4		
GP14_5	B37	262				GP14_5	ICU5 TIN5/13 TTG29/21/13/5		
GP14_6	A37	117				GP14_6	ICU6 TIN6/14 TTG30/22/14/6		
GP14_7	C34	402				GP14_7	ICU7 TIN7/15 TTG31/23/15/7 STOPWT		
GP15_0	F30	769	I/O	A	VDD5	GP15_0	OCU0	TOT0	
GP15_1	B35	264				GP15_1	OCU1	TOT1	
GP15_2	C33	403				GP15_2	OCU2	TOT2	
GP15_3	E30	656				GP15_3	OCU3	TOT3	
GP15_4	A35	119				GP15_4	OCU4	TOT4	
GP15_5	E29	657				GP15_5	OCU5	TOT5	
GP15_6	D30	535				GP15_6	OCU6	TOT6	
GP15_7	D28	537				GP15_7	OCU7	TOT7	

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP16_0	F28	771	I/O	B1	VDD5	GP16_0	PPG8		
GP16_1	A34	120				GP16_1	PPG9		
GP16_2	C31	405		A		GP16_2	PPG10		
GP16_3	E28	658				GP16_3	PPG11		
GP16_4	D29	536				GP16_4	PPG12	SGA	
GP16_5	A33	121				GP16_5	PPG13	SGO	
GP16_6	C29	407				GP16_6	PPG14	PFM	
GP16_7	E26	660				GP16_7	PPG15	ATGX0/1	
GP17_0	AT11	472	I/O	B1	VDD5	GP17_0	PPG0	RCK1	AN32 *3
GP17_1	AV8	195				GP17_1	PPG1	RDA10	AN33 *3
GP17_2	AP12	715				GP17_2	PPG2	RDA11	AN34 *3
GP17_3	AT12	473				GP17_3	PPG3		AN35 *3
GP17_4	AT10	471				GP17_4	PPG4	TCK1	AN36 *3
GP17_5	AR11	597				GP17_5	PPG5	TDA10	AN37 *3
GP17_6	AU8	336				GP17_6	PPG6	TDA11	AN38 *3
GP17_7	AR10	596				GP17_7	PPG7	TEN1	AN39 *3
GP18_0	AT8	469	I/O	B1	VDD5	GP18_0	SIN6	AIN2	AN40 *3
GP18_1	AU6	334				GP18_1	SOT6	BIN2	AN41 *3
GP18_2	AW3	41				GP18_2	SCK6	ZIN2 CK6	AN42 *3
GP18_3	AR8	594				GP18_3			AN43 *3
GP18_4	AR7	593				GP18_4	SIN7	AIN3	AN44 *3
GP18_5	AU2	188				GP18_5	SOT7	BIN3	AN45 *3
GP18_6	AM5	588				GP18_6	SCK7	ZIN3 CK7	AN46 *3
GP18_7	AL5	587				GP18_7			AN47 *3
GP19_0	A31	123	I/O	A	VDD5	GP19_0	SIN4		
GP19_1	C28	408				GP19_1	SOT4		
GP19_2	D26	539		B1		GP19_2	SCK4	CK4	
GP19_3	AK6	705				GP19_3			AN48 *3
GP19_4	B29	270		A		GP19_4	SIN5		
GP19_5	A30	124				GP19_5	SOT5		
GP19_6	C26	410				GP19_6	SCK5	CK5	
GP19_7	AP3	328		B1		GP19_7			AN49 *3

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP20_0	F26	773	I/O	B1	VDD5	GP20_0	SIN2	AIN0	
GP20_1	D27	538				GP20_1	SOT2	BIN0	
GP20_2	B30	269				GP20_2	SCK2	ZIN0 CK2/10	
GP20_3	AJ5	585				GP20_3			AN50 *3
GP20_4	E27	659				GP20_4	SIN3	AIN1	
GP20_5	C30	406				GP20_5	SOT3	BIN1	
GP20_6	B32	267				GP20_6	SCK3	ZIN1 CK3/11	
GP20_7	AM3	326				GP20_7			AN51 *3
GP21_0	A29	125	I/O	A	VDD5	GP21_0	SIN0		
GP21_1	B27	272				GP21_1	SOT0		
GP21_2	E24	662		B1		GP21_2	SCK0	CK0/8	
GP21_3	AG5	583				GP21_3			AN52 *3
GP21_4	F24	775		A		GP21_4	SIN1		
GP21_5	C24	412				GP21_5	SOT1		
GP21_6	A27	127		B1		GP21_6	SCK1	CK1/9	
GP21_7	AK2	181				GP21_7			AN53 *3
GP22_0	D23	542	I/O	A	VDD5	GP22_0	RX4 INT12		
GP22_1	E23	663				GP22_1	TX4		
GP22_2	B24	275				GP22_2	RX5 INT13		
GP22_3	E22	664				GP22_3	TX5		
GP22_4	A24	130		C		GP22_4	SDA0 INT14		
GP22_5	D21	544				GP22_5	SCL0		
GP22_6	B20	279				GP22_6 ICU8 *4	SDA1 INT15		
GP22_7	C20	416				GP22_7 ICU9 *4	SCL1		

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP23_0	A20	134	I/O	A	VDD5	GP23_0	RX0 INT8		
GP23_1	F20	779				GP23_1	TX0		
GP23_2	A18	136				GP23_2	RX1 INT9		
GP23_3	B18	281				GP23_3	TX1		
GP23_4	C19	417				GP23_4	RX2 INT10		
GP23_5	D19	546				GP23_5	TX2		
GP23_6	E19	667				GP23_6	RX3 INT11		
GP23_7	D18	547				GP23_7	TX3		
GP24_0	C32	404	I/O	B1	VDD5	GP24_0	INT0		
GP24_1	B34	265				GP24_1	INT1		
GP24_2	A36	118				GP24_2	INT2		
GP24_3	D32	533				GP24_3	INT3		
GP24_4	E31	655		D		GP24_4	SDA2 INT4		
GP24_5	B36	263				GP24_5	SCL2 INT5		
GP24_6	D34	531				GP24_6	SDA3 INT6		
GP24_7	F32	767				GP24_7	SCL3 INT7		
GP25_0	B25	274	I/O	E	HVDD	GP25_0	SMC1P4		
GP25_1	A26	128				GP25_1	SMC1M4		
GP25_2	D24	541				GP25_2	SMC2P4		
GP25_3	A28	126				GP25_3	SMC2M4		
GP25_4	C25	411				GP25_4	SMC1P5		
GP25_5	B26	273				GP25_5	SMC1M5		
GP25_6	D25	540				GP25_6	SMC2P5		
GP25_7	B28	271				GP25_7	SMC2M5		

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP26_0	A19	135	I/O	F	HVDD	GP26_0	SMC1P2	AN24	INT24 *5
GP26_1	E20	666				GP26_1	SMC1M2	AN25	INT25 *5
GP26_2	D20	545				GP26_2	SMC2P2	AN26	INT26 *5
GP26_3	A21	133				GP26_3	SMC2M2	AN27	INT27 *5
GP26_4	B22	277				GP26_4	SMC1P3	AN28	INT28 *5
GP26_5	C21	415				GP26_5	SMC1M3	AN29	INT29 *5
GP26_6	B21	278				GP26_6	SMC2P3	AN30	INT30 *5
GP26_7	C22	414				GP26_7	SMC2M3	AN31	INT31 *5
GP27_0	B14	285	I/O	F	HVDD	GP27_0	SMC1P0	AN16	INT16 *5
GP27_1	A12	142				GP27_1	SMC1M0	AN17	INT17 *5
GP27_2	E17	669				GP27_2	SMC2P0	AN18	INT18 *5
GP27_3	A15	139				GP27_3	SMC2M0	AN19	INT19 *5
GP27_4	C18	418				GP27_4	SMC1P1	AN20	INT20 *5
GP27_5	A16	138				GP27_5	SMC1M1	AN21	INT21 *5
GP27_6	B17	282				GP27_6	SMC2P1	AN22	INT22 *5
GP27_7	B19	280				GP27_7	SMC2M1	AN23	INT23 *5
GP28_0	B9	290	I/O	B1	VDD5	GP28_0	AN8	RCK0	
GP28_1	E13	673				GP28_1	AN9	RDA00	
GP28_2	A6	148				GP28_2	AN10	RDA01	
GP28_3	B7	292				GP28_3	AN11		
GP28_4	C9	427				GP28_4	AN12	TCK0	
GP28_5	E10	676				GP28_5	AN13	TDA00	
GP28_6	B5	294		B2		GP28_6	AN14 DA0 *6	TDA01	
GP28_7	D8	557				GP28_7	AN15 DA1 *6	TEN0	
GP29_0	B10	289	I/O	B1	VDD5	GP29_0	AN0		
GP29_1	A8	146				GP29_1	AN1		
GP29_2	C12	424				GP29_2	AN2		
GP29_3	B12	287				GP29_3	AN3		
GP29_4	A10	144				GP29_4	AN4		
GP29_5	C15	421				GP29_5	AN5		
GP29_6	E16	670				GP29_6	AN6		
GP29_7	C14	422				GP29_7	AN7		

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP30_0	AD5	580	I/O	K	VDD5	GP30_0	COM0	PPG16	
GP30_1	AJ1	29				GP30_1	COM1	PPG17	
GP30_2	AK1	30				GP30_2	COM2	PPG18	
GP30_3	AH2	179				GP30_3	COM3	PPG19	
GP30_4	AJ2	180		L		GP30_4	V0	PPG20	
GP30_5	AF4	455				GP30_5	V1	PPG21	
GP30_6	AM1	32				GP30_6	V2	PPG22	
GP30_7	AL1	31				GP30_7	V3	PPG23	
GP31_0	AD6	699	I/O	K	VDD5	GP31_0	SEG32	TXDA	
GP31_1	AH1	28				GP31_1	SEG33	TXENA	
GP31_2	AE1	25				GP31_2	SEG34	RXDA	
GP31_3	AB4	451				GP31_3	SEG35		
GP31_4	AB3	316				GP31_4	SEG36	TXDB	
GP31_5	AC2	174				GP31_5	SEG37	TXENB	
GP31_6	AA2	172				GP31_6	SEG38	RXDB	
GP31_7	Y5	576				GP31_7	SEG39		
GP32_0	Y4	449	I/O	K	VDD5	GP32_0	SEG24	SIN14	
GP32_1	W1	19				GP32_1	SEG25	SOT14	
GP32_2	V1	18				GP32_2	SEG26	SCK14	
GP32_3	W5	575				GP32_3	SEG27	PPG30	
GP32_4	U1	17				GP32_4	SEG28	SIN15	
GP32_5	T1	16				GP32_5	SEG29	SOT15	
GP32_6	W3	313				GP32_6	SEG30	SCK15	
GP32_7	V5	574				GP32_7	SEG31	PPG31	
GP33_0	R2	166	I/O	K	VDD5	GP33_0	SEG16	SIN12	
GP33_1	U5	573				GP33_1	SEG17	SOT12	
GP33_2	P1	14				GP33_2	SEG18	SCK12	
GP33_3	T4	445				GP33_3	SEG19	PPG28	
GP33_4	T5	572				GP33_4	SEG20	SIN13	
GP33_5	P2	165				GP33_5	SEG21	SOT13	
GP33_6	R5	571				GP33_6	SEG22	SCK13	
GP33_7	M2	163				GP33_7	SEG23	PPG29	

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP34_0	L2	162	I/O	K	VDD5	GP34_0	SEG8	SIN10	
GP34_1	P6	689				GP34_1	SEG9	SOT10	
GP34_2	J1	9				GP34_2	SEG10	SCK10	
GP34_3	L3	305				GP34_3	SEG11	PPG26	
GP34_4	J2	160				GP34_4	SEG12	SIN11	
GP34_5	L4	440				GP34_5	SEG13	SOT11	
GP34_6	J3	303				GP34_6	SEG14	SCK11	
GP34_7	M6	687				GP34_7	SEG15	PPG27	
GP35_0	G2	158	I/O	K1	VDD5	GP35_0	SEG0	SIN8	
GP35_1	K4	439				GP35_1	SEG1	SOT8	
GP35_2	E1	5		K		GP35_2	SEG2	SCK8	
GP35_3	F2	157				GP35_3	SEG3	PPG24	
GP35_4	E2	156		K1		GP35_4	SEG4	SIN9	
GP35_5	K6	685				GP35_5	SEG5	SOT9	
GP35_6	F3	300		K		GP35_6	SEG6	SCK9	
GP35_7	C2	154				GP35_7	SEG7	PPG25	
GP36_0	AR30	616	I/O	R	VDDM	GP36_0			
GP36_1	AR29	615				GP36_1			
GP36_2	AU31	359				GP36_2	WS0		
GP36_3	AV33	220				GP36_3	ISCK0		
GP36_4	AW34	72				GP36_4			
GP36_5	AR27	613		S		GP36_5	MLBDAT		
GP36_6	AV30	217				GP36_6	MLBSIG		
GP36_7	AW32	70				GP36_7	MLBCLK		
GP37_0	F18	781	I/O	A	VDD5	GP37_0			
GP37_1	B16	283				GP37_1			
GP37_2	C17	419				GP37_2			
GP37_3	C16	420				GP37_3			
GP37_4	A13	141				GP37_4			
GP37_5	B13	286				GP37_5			
GP37_6	D15	550				GP37_6			
GP37_7	F16	783				GP37_7			

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	PFR=0 EPFR=0	PFR=1 EPFR=0	PFR=1 EPFR=1	Special settings or Comments
GP38_0	AT26	487	I/O	R	VDDM	GP38_0	SD8		
GP38_1	AW30	68				GP38_1	SD9		
GP38_2	AR24	610				GP38_2			
GP38_3	AP24	727				GP38_3			
GP38_4	AT25	486				GP38_4			
GP38_5	AU24	352				GP38_5			
GP38_6	AR23	609				GP38_6			
GP38_7	AW25	63				GP38_7			
GP39_0	AR22	608	I/O	R	VDDM	GP39_0	SD0		
GP39_1	AT22	483				GP39_1	SD1		
GP39_2	AV23	210				GP39_2	SD2		
GP39_3	AV21	208				GP39_3	SD3		
GP39_4	AW19	57				GP39_4	SD4		
GP39_5	AU20	348				GP39_5	SD5		
GP39_6	AV20	207				GP39_6	SD6		
GP39_7	AW17	55				GP39_7	SD7		
GP40_0	D16	549	I/O	C	VDD5	GP40_0	SDA4		
GP40_1	A11	143				GP40_1	SCL4		
GP40_2	E15	671				GP40_2	SDA5		
GP40_3	B11	288				GP40_3	SCL5		
GP40_4	C13	423				GP40_4	SDA6		
GP40_5	D13	552				GP40_5	SCL6		
GP40_6	E14	672				GP40_6	SDA7		
GP40_7	F14	785				GP40_7	SCL7		

1. About the types of I/O cells, please refer to chapter “I/O Circuit Types MB91FV460B” on page 179.
2. About the I/O power domains, please refer to section “Power Supply Pins” on page 175.
3. AN32 to AN53 are enabled by the ADC channel enable bits in AD1ERH/AD1ERL registers. Setting these bits will disable the digital input lines of the port. See the description in section 44.4. Analog Input Connections (Page No.1054).
4. ICU8, ICU9: If PFR is set, ICU8/9 are connected to LIN-USART8/9 (LSYNC). If PFR is cleared, ICU8/9 work as normal ICU.
5. For usage of INT16 to INT31, the ADC channel must be disabled (channel enable bit in ADERH/ADERL is cleared) and the SMC analog function must be disabled. Setting the interrupt enable bit in ENIR0, ENIR1 registers will enable wakeup in STOP mode.
6. The D/A Converter outputs DA0, DA1 are enabled by setting DAE0,DAE1 in DACR register. The ADC channel can be enabled in parallel to work on the same analog line.

3.8.4 Multiplexed Pins in Series Mode P/T

On MB91460P and MB91460T series, the pin multiplexing is controlled by PPMUX register settings. Please refer to the datasheet of MB91460P or MB91460T series for the explanation of the PPMUX registers.

If pins are multiplexed, the port control registers (PDR, DDR, PFR, EPFR, ...) of the multiplex source port are used and the input lines of the multiplex destination port are clipped to 1.

For example, if GP24_0/INT0 are multiplexed to the pin of GP00_0, then the control registers of GP24_0 are to be used and the input lines of GP00_0 are clipped to 1.

Note: Don't set port function registers for resources, which are not available on MB91460P/T series. For example, setting PFR24[7:4]=1 && EPFR24[7:4]=1 will switch I2C signals to GP00[7:4]. But Port GP00 does not have I2C I/O cells and the I2C modules are not available on MB91460P/T series.

Table 3.8-4 Multiplexed pins in series mode P/T

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Pin multiplexing	PPMUX register bit	PPMUX2 reg. bit
GP00_0	AP35	622	I/O	A	VDD35	GP24_0 / INT0	PR0=1	-
GP00_1	AR36	498				GP24_1 / INT1		
GP00_2	AT37	366				GP24_2 / INT2		
GP00_3	AU38	226				GP24_3 / INT3		
GP00_4	AL35	625				GP24_4 / INT4		
GP00_5	AK34	741				GP24_5 / INT5		
GP00_6	AR38	228				GP24_6 / INT6		
GP00_7	AN37	369				GP24_7 / INT7		
GP01_0	AL36	502	I/O	A	VDD35	GP17_0 / PPG0	PS3=1	-
GP01_1	AM37	370				GP17_1 / PPG1		
GP01_2	AP39	82				GP17_2 / PPG2		
GP01_3	AL37	371				GP17_3 / PPG3		
GP01_4	AK37	372				GP15_4 / OCU4 / TOT4		
GP01_5	AL38	232				GP15_5 / OCU5 / TOT5		
GP01_6	AG36	506				GP15_6 / OCU6 / TOT6		
GP01_7	AF35	630				GP15_7 / OCU7 / TOT7		

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Pin multiplexing	PPMUX register bit	PPMUX2 reg. bit
GP05_0	K36	523	I/O	A	VDD35	GP16_0 / PPG8	PR10=1	-
GP05_1	D39	112				GP16_1 / PPG9	PR11=1	
GP05_2	E38	258				GP20_0 / SIN2 / AIN0 or GP34_0 / SIN10	PR12=1 && PRPS0=1 PR12=1 && PRPS0=0	PR0=0
GP05_3	J35	647				GP20_1 / SOT2 / BIN0 or GP34_1 / SOT10	PR13=1 && PRPS0=1 PR13=1 && PRPS0=0	PR1=0
GP05_4	H35	648				GP20_2 / SCK2 / ZIN0 or GP34_2 / SCK10	PR14=1 && PRPS0=1 PR14=1 && PRPS0=0	PR2=0
GP05_5	D38	259				GP20_4 / SIN3 / AIN1 or GP34_4 / SIN11	PR15=1 && PRPS0=1 PR15=1 && PRPS0=0	PR3=0
GP05_6	E37	397				GP20_5 / SOT3 / BIN1 or GP34_5 / SOT11	PR16=1 && PRPS0=1 PR16=1 && PRPS0=0	PR4=0
GP05_7	E36	528				GP20_6 / SCK3 / ZIN1 or GP34_6 / SCK11	PR17=1 && PRPS0=1 PR17=1 && PRPS0=0	PR5=0
GP06_0	T35	640	I/O	B1	VDD35	GP21_0 / SIN0	PS4=1	-
GP06_1	R35	641				GP21_1 / SOT0		
GP06_2	L38	252				GP21_2 / SCK0 / CK0/8		
GP06_3	N36	520				GP17_4 / PPG4		
GP06_4	P35	642				GP14_4 / ICU4 / TIN4/12 / TTG28/20/12/4		
GP06_5	J39	107				GP14_5 / ICU5 / TIN5/13 / TTG29/21/13/5		
GP06_6	N35	643				GP14_6 / ICU6 / TIN6/14 / TTG30/22/14/6		
GP06_7	G39	109				GP14_7 / ICU7 / TIN7/15 / TTG31/23/15/7		

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Pin multiplexing	PPMUX register bit	PPMUX2 reg. bit
GP07_0	M35	644	I/O	B1	VDD35	GP26_0 / AN24	PS0=1	-
GP07_1	J37	393				GP26_1 / AN25		
GP07_2	L35	645				GP26_2 / AN26		
GP07_3	K35	646				GP26_3 / AN27		
GP07_4	F37	396				GP26_4 / AN28		
GP07_5	K34	761				GP26_5 / AN29		
GP07_6	G36	526				GP26_6 / AN30		
GP07_7	C38	260				GP26_7 / AN31		
GP09_0	AJ39	87	I/O	A	VDD35	GP34_4 / SIN11	-	PR3=1
GP09_1	AH39	88				GP34_5 / SOT11	-	PR4=1
GP09_2	AD36	509				GP34_6 / SCK11	-	PR5=1
GP10_0	AB35	634	I/O	A	VDD35	GP34_0 / SIN10	-	PR0=1
GP10_1	AD39	92				GP34_1 / SOT10	-	PR1=1
GP10_3	AB38	241				GP34_2 / SCK10	-	PR2=1
GP16_0	F28	771	I/O	B1	VDD5	GP27_0 / AN16	PS1=1 PR10=1	-
GP16_1	A34	120				GP27_1 / AN17	PS1=1 PR11=1	-
GP20_0	F26	773	I/O	B1	VDD5	GP27_2 / AN18	PS1=1 (PR12=1 && PRPS0=1)	-
GP20_1	D27	538				GP27_3 / AN19	PS1=1 (PR13=1 && PRPS0=1)	-
GP20_2	B30	269				GP27_4 / AN20	PS1=1 (PR14=1 && PRPS0=1)	-
GP20_4	E27	659				GP27_5 / AN21	PS1=1 (PR15=1 && PRPS0=1)	-
GP20_5	C30	406				GP27_6 / AN22	PS1=1 (PR16=1 && PRPS0=1)	-
GP20_6	B32	267				GP27_7 / AN23	PS1=1 (PR17=1 && PRPS0=1)	-

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Pin multiplexing	PPMUX register bit	PPMUX2 reg. bit
GP24_0	C32	404	I/O	B1	VDD5	GP28_0 / AN8	PS2=1 PR0=1	-
GP24_1	B34	265				GP28_1 / AN9		
GP24_2	A36	118				GP28_2 / AN10		
GP24_3	D32	533				GP28_3 / AN11		
GP24_4	E31	655		D		GP28_4 / AN12		
GP24_5	B36	263				GP28_5 / AN13		
GP24_6	D34	531				GP28_6 / AN14		
GP24_7	F32	767				GP28_7 / AN15		
GP35_0	G2	158	I/O	K1	VDD5	GP29_0 / AN0	PS5=1	-
GP35_1	K4	439				GP29_1 / AN1		
GP35_2	E1	5				GP29_2 / AN2		
GP35_4	E2	156	I/O	K1		GP29_3 / AN3		
GP35_5	K6	685				GP29_4 / AN4		
GP35_6	F3	300				GP29_5 / AN5		

1. About the types of I/O cells, please refer to chapter “[I/O Circuit Types MB91FV460B](#)” on page 179.
2. About the I/O power domains, please refer to section “[Power Supply Pins](#)” on page 175.

3.8.5 Multiplexed Pins in Series Mode B

The pin multiplexing on MB91460B series is controlled by MD_3 pin. The following table lists the multiplexed functions in “No External Bus” mode, which is enabled by setting MD_3 = 0 on MB91460B series devices.

If pins are multiplexed, the port control registers (PDR, DDR, PFR, EPFR, ...) of the multiplex source port are used and the input lines of the multiplex destination port are clipped to 1.

Note: Don't set port function registers for resources, which are not available on MB91460B series. For example, setting PFR24[7:4]=1 && EPFR24[7:4]=1 will switch I2C signals to GP05[7:4]. But Port GP05 does not have I2C I/O cells and the I2C modules are not available on MB91460B series.

Table 3.8-5 Multiplexed pins in series mode B

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Series Mode B Pin Multiplexing
GP00_0	AP35	622	I/O	A	VDD35	GP17_0 / PPG0
GP00_1	AR36	498				GP17_1 / PPG1
GP00_2	AT37	366				GP17_2 / PPG2
GP00_3	AU38	226				GP17_3 / PPG3
GP00_4	AL35	625				GP17_4 / PPG4
GP00_5	AK34	741				GP17_5 / PPG5
GP00_6	AR38	228				GP17_6 / PPG6
GP00_7	AN37	369				GP17_7 / PPG7
GP01_0	AL36	502	I/O	A	VDD35	GP14_4 / ICU4 / TIN4/12 / TTG28/20/12/4
GP01_1	AM37	370				GP14_5 / ICU5 / TIN5/13 / TTG29/21/13/5
GP01_2	AP39	82				GP14_6 / ICU6 / TIN6/14 / TTG30/22/14/6
GP01_3	AL37	371				GP14_7 / ICU7 / TIN7/15 / TTG31/23/15/7
GP01_4	AK37	372				GP15_4 / OCU4 / TOT4
GP01_5	AL38	232				GP15_5 / OCU5 / TOT5
GP01_6	AG36	506				GP15_6 / OCU6 / TOT6
GP01_7	AF35	630				GP15_7 / OCU7 / TOT7
GP05_0	K36	523	I/O	A	VDD35	GP24_4 / INT4
GP05_1	D39	112				GP24_5 / INT5
GP05_2	E38	258				GP24_6 / INT6
GP05_3	J35	647				GP24_7 / INT7
GP05_4	H35	648				GP21_0 / SIN0
GP05_5	D38	259				GP21_1 / SOT0

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Series Mode B Pin Multiplexing
GP06_0	T35	640	I/O	B1	VDD35	GP26_0 / AN24
GP06_1	R35	641				GP26_1 / AN25
GP06_2	L38	252				GP26_2 / AN26
GP06_3	N36	520				GP26_3 / AN27
GP06_4	P35	642				GP26_4 / AN28
GP06_5	J39	107				GP26_5 / AN29
GP06_6	N35	643				GP26_6 / AN30
GP06_7	G39	109				GP26_7 / AN31
GP07_0	M35	644	I/O	B1	VDD35	GP27_0 / AN16
GP07_1	J37	393				GP27_1 / AN17
GP07_2	L35	645				GP27_2 / AN18
GP07_3	K35	646				GP27_3 / AN19
GP07_4	F37	396				GP27_4 / AN20
GP07_5	K34	761				GP27_5 / AN21
GP07_6	G36	526				GP27_6 / AN22
GP07_7	C38	260				GP27_7 / AN23
GP08_0	AG37	375	I/O	A	VDD35	GP20_4 / SIN3 / AIN1
GP08_1	AH38	235				GP24_0 / INT0
GP08_4	AD37	378				GP20_5 / SOT3 / BIN1
GP08_7	AE39	91				GP20_6 / SCK3 / ZIN1 / CK3/11
GP09_0	AJ39	87				GP20_1 / SOT2 / BIN0
GP09_1	AH39	88				GP20_2 / SCK2 / ZIN0 / CK2/10
GP10_0	AB35	634				GP20_0 / SIN2 / AIN0

1. About the types of I/O cells, please refer to chapter “I/O Circuit Types MB91FV460B” on page 179.

2. About the I/O power domains, please refer to section “Power Supply Pins” on page 175.

3.8.6 Relocated External Interrupts in Series Mode Q

On MB91460Q series, the following external interrupt lines are permanently moved to other pins.
For information about how to enable these interrupts and wakeup from STOP mode, please refer to section [25.4.6 Interrupt Re-location \(Page No.456\)](#).

Table 3.8-6 Relocated external interrupts in series mode Q

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Relocated Interrupts in Series Mode Q
GP24_4	E31	655	I/O	C	VDD5	INT14-Q
GP34_0	L2	162	I/O	K		INT10-Q
GP34_4	J2	160	I/O	K		INT4-Q
GP34_5	L4	440	I/O	K		INT5-Q
GP35_0	G2	158	I/O	K1		INT11-Q
GP35_4	E2	156	I/O	K1		INT12-Q

1. About the types of I/O cells, please refer to chapter [3.9. I/O Circuit Types MB91FV460B \(Page No.179\)](#).
2. About the I/O power domains, please refer to section [3.8.9 Power Supply Pins \(Page No.175\)](#).

3.8.7 Relocated External Interrupts in Series Mode M

On MB91460M series, the following external interrupt lines are permanently moved to other pins.
For information about how to enable these interrupts and wakeup from STOP mode, please refer to section [25.4.6 Interrupt Re-location \(Page No.456\)](#).

Pin Name	JEDEC	Pin no.	I/O	I/O Circuit type *1	Power domain *2	Relocated Interrupts in Series Mode M
GP22_1	E23	663	I/O	A	VDD5	INT14-M
GP22_3	E22	664	I/O	A	VDD5	INT15-M

1. About the types of I/O cells, please refer to chapter [3.9. I/O Circuit Types MB91FV460B \(Page No.179\)](#).
2. About the I/O power domains, please refer to section [3.8.9 Power Supply Pins \(Page No.175\)](#).

MB91460 Series

3.8.8 Dedicated and Special Pins

The following table lists the dedicated (non-multiplexed) and special pins of MB91FV460B:

Table 3.8-7 Dedicated and special pins

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
ALARM_1	C3	297	I	N1	VDD5	ALARM comparator 1 input	
ALARM_0	H6	683	I	N1	VDD5	ALARM comparator 0 input	
HCONX	AU35	363	O	U2	VDDU	USB external transistor control	during INITX=0, high level is output
UDP	AU33	361	I/O	Q		USB I/O	
UDM	AP30	733	I/O	Q		USB I/O	
VSSA_1	AU19	347	I	Z0	VDDA	APIX VSS	
SDINP	AV19	206	I	N1		APIX data in	
SDINM	AR18	604	I	N1		APIX data in	
SDOUTP	AT18	479	O	N2		APIX data out	
SDOUTM	AW15	53	O	N2		APIX data out	
VDDA	AR17	603	I	Z1		APIX VDD	
VSSA_2	AW13	51	I	Z0		APIX VSS	
X1	AU14	342	O	J1	VDD5	4 MHz main oscillator output	
X0	AT15	476	I			4 MHz main oscillator input	
X1A	AU13	341	O	J2		32 kHz sub oscillator output	
X0A	AW10	48	I			32 kHz sub oscillator input	
MONCLK	AP16	719	M	M	VDD5	Clock monitor output	3-state output
WDRESX	F8	791	O	O	VDD5	Hardware Watchdog reset output	
INITX	C11	425	I	H1	VDD5	External reset input	Pull-Up
RSTX	C10	426	I	H1		External software reset input	Pull-Up
HSTX	A9	145	I	H1		Hardware standby	Pull-Up
NMIX	D12	553	I	H1		Non-maskable interrupt	Pull-Up
FCI	D11	554	I	H3		Fast clock input enable	Pull-down
MD_2	AW8	46	I	G	VDD5	Mode setting pin	
MD_1	AW9	47	I	G		Mode setting pin	
MD_0	AU12	340	I	G		Mode setting pin	
SER_MD_2	E11	675	I	H3	VDD5	Series emulation mode	Pull-down
SER_MD_1	C7	429	I	H3		Series emulation mode	Pull-down
SER_MD_0	C6	430	I	H3		Series emulation mode	Pull-down
COMPAT	C8	428	I	H3		Test pin, keep input low	Pull-down

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
SRAM_SFX	D9	556	I	H1	VDD5	External S-RAM / Flash connected	Pull-Up
FSC_DISABLE	E12	674	I	H1		Flash Security Controller disable	Pull-Up
CSV_KILL	F12	787	I	H1		Clock Supervisor disable	Pull-Up
HWWDG_KILL	A7	147	I	H1		Hardware Watchdog disable	Pull-Up
FIX_ENX	B8	291	I	H1		Fixed Reset Vevtor enable	Pull-Up
EDSU_BREAKX	E9	677	I	H1		BREAK input for EDSU	Pull-Up
ICLK	F6	681	O	W	VDD3	DSU4 ICE interface clock output	input in SCAN test mode only
ICD_3	F5	562	I/O	T2		DSU4 ICE interface data	per default, Pull-down is enabled
ICD_2	D4	433	I/O	T2		DSU4 ICE interface data	
ICD_1	D3	298	I/O	T2		DSU4 ICE interface data	
ICD_0	G5	563	I/O	T2		DSU4 ICE interface data	
ICS_2	H5	564	O	U1		DSU4 ICE interface state	during TRSTX = 0, low level is output
ICS_1	E3	299	O	U1		DSU4 ICE interface state	
ICS_0	E5	561	O	U1		DSU4 ICE interface state	
BREAK	D2	155	I	T3	VDD3	DSU4 Control	per default, Pull-down is enabled
PLEVEL	C1	3	I	T4		DSU4 Control	Pull-Down
EXRAM	AR37	367	I	T1		DSU4 Control	
TRSTX	G4	436	I	TR		DSU4 Control	
TCLK	J5	565	O	V	VDD3	DSU4 Trace RAM clock	
TOEX	D1	4	O	U0		DSU4 Trace RAM interface control	
TWRX	T6	691				DSU4 Trace RAM interface control	
TCE1X	H4	437				DSU4 Trace RAM interface control	
TADSCX	K5	566				DSU4 Trace RAM interface control	
TAD_15	AK5	586	O	U0	VDD3	DSU4 Trace RAM Address	
TAD_14	AR1	35				DSU4 Trace RAM Address	
TAD_13	AT1	36				DSU4 Trace RAM Address	
TAD_12	AP2	185				DSU4 Trace RAM Address	
TAD_11	AR2	186				DSU4 Trace RAM Address	
TAD_10	AN3	327				DSU4 Trace RAM Address	
TAD_9	AM4	461				DSU4 Trace RAM Address	
TAD_8	AN4	462				DSU4 Trace RAM Address	

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
TAD_7	AU1	37	O	U0	VDD3	DSU4 Trace RAM Address	
TAD_6	AT2	187				DSU4 Trace RAM Address	
TAD_5	AR3	329				DSU4 Trace RAM Address	
TAD_4	AT3	330				DSU4 Trace RAM Address	
TAD_3	AN5	589				DSU4 Trace RAM Address	
TAD_2	AR4	464				DSU4 Trace RAM Address	
TAD_1	AM6	707				DSU4 Trace RAM Address	
TAD_0	AP5	590				DSU4 Trace RAM Address	
TDT_68	P4	443	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_67	N3	307				DSU4 Trace RAM Data	
TDT_66	K1	10				DSU4 Trace RAM Data	
TDT_65	R4	444				DSU4 Trace RAM Data	
TDT_64	P3	308				DSU4 Trace RAM Data	
TDT_63	N2	164	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_62	R3	309				DSU4 Trace RAM Data	
TDT_61	M1	12				DSU4 Trace RAM Data	
TDT_60	T3	310				DSU4 Trace RAM Data	
TDT_59	N1	13				DSU4 Trace RAM Data	
TDT_58	U3	311				DSU4 Trace RAM Data	
TDT_57	U4	446				DSU4 Trace RAM Data	
TDT_56	V3	312				DSU4 Trace RAM Data	
TDT_55	R1	15	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_54	V4	447				DSU4 Trace RAM Data	
TDT_53	W2	170				DSU4 Trace RAM Data	
TDT_52	V6	693				DSU4 Trace RAM Data	
TDT_51	W4	448				DSU4 Trace RAM Data	
TDT_50	U2	168				DSU4 Trace RAM Data	
TDT_49	V2	169				DSU4 Trace RAM Data	
TDT_48	Y2	171				DSU4 Trace RAM Data	

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
TDT_47	Y3	314	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_46	Y1	20				DSU4 Trace RAM Data	
TDT_45	AA1	21				DSU4 Trace RAM Data	
TDT_44	Y6	695				DSU4 Trace RAM Data	
TDT_43	AB1	22				DSU4 Trace RAM Data	
TDT_42	AB2	173				DSU4 Trace RAM Data	
TDT_41	AC1	23				DSU4 Trace RAM Data	
TDT_40	AA3	315				DSU4 Trace RAM Data	
TDT_39	AA4	450	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_38	AA5	577				DSU4 Trace RAM Data	
TDT_37	AD1	24				DSU4 Trace RAM Data	
TDT_36	AB6	697				DSU4 Trace RAM Data	
TDT_35	AB5	578				DSU4 Trace RAM Data	
TDT_34	AD2	175				DSU4 Trace RAM Data	
TDT_33	AC3	317				DSU4 Trace RAM Data	
TDT_32	AE2	176				DSU4 Trace RAM Data	
TDT_31	AC4	452	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_30	AF1	26				DSU4 Trace RAM Data	
TDT_29	AD3	318				DSU4 Trace RAM Data	
TDT_28	AG1	27				DSU4 Trace RAM Data	
TDT_27	AF2	177				DSU4 Trace RAM Data	
TDT_26	AG2	178				DSU4 Trace RAM Data	
TDT_25	AF3	320				DSU4 Trace RAM Data	
TDT_24	AD4	453				DSU4 Trace RAM Data	
TDT_23	AE3	319	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_22	AE5	581				DSU4 Trace RAM Data	
TDT_21	AG3	321				DSU4 Trace RAM Data	
TDT_20	AH3	322				DSU4 Trace RAM Data	
TDT_19	AG4	456				DSU4 Trace RAM Data	
TDT_18	AF5	582				DSU4 Trace RAM Data	
TDT_17	AF6	701				DSU4 Trace RAM Data	
TDT_16	AL2	182				DSU4 Trace RAM Data	

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
TDT_15	AJ3	323	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_14	AK3	324				DSU4 Trace RAM Data	
TDT_13	AJ4	458				DSU4 Trace RAM Data	
TDT_12	AH5	584				DSU4 Trace RAM Data	
TDT_11	AP1	34				DSU4 Trace RAM Data	
TDT_10	AH6	703				DSU4 Trace RAM Data	
TDT_9	AL3	325				DSU4 Trace RAM Data	
TDT_8	AR5	591				DSU4 Trace RAM Data	
TDT_7	AL4	460	I/O	X	VDD3	DSU4 Trace RAM Data	
TDT_6	AR6	592				DSU4 Trace RAM Data	
TDT_5	AT4	465				DSU4 Trace RAM Data	
TDT_4	AP8	711				DSU4 Trace RAM Data	
TDT_3	AU4	332				DSU4 Trace RAM Data	
TDT_2	AT6	467				DSU4 Trace RAM Data	
TDT_1	AV3	190				DSU4 Trace RAM Data	
TDT_0	AV4	191				DSU4 Trace RAM Data	
EMRAM	AN35	623	I	T1	VDD3	External S-RAM / Flash control	
ECSX	AU39	79	O	U1	VDD3	External S-RAM / Flash control	
EWR3X	AM34	739				External S-RAM / Flash control	
EWR2X	AU37	365				External S-RAM / Flash control	
EWR1X	AT36	497				External S-RAM / Flash control	
EWR0X	AP34	737				External S-RAM / Flash control	
EECSX	AT38	227				External S-RAM / Flash control	
EEWEX	AR34	620				External S-RAM / Flash control	
EEOEX	AT35	496				External S-RAM / Flash control	
EEBEX_7	AP32	735	O	U1	VDD3	External S-RAM / Flash byte enable	
EEBEX_6	AR33	619				External S-RAM / Flash byte enable	
EEBEX_5	AU36	364				External S-RAM / Flash byte enable	
EEBEX_4	AT34	495				External S-RAM / Flash byte enable	
EEBEX_3	AV37	224				External S-RAM / Flash byte enable	
EEBEX_2	AV36	223				External S-RAM / Flash byte enable	
EEBEX_1	AR32	618				External S-RAM / Flash byte enable	
EEBEX_0	AT33	494				External S-RAM / Flash byte enable	

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
EEA_19	AU27	355	O	U0	VDD3	External S-RAM / Flash address	
EEA_18	AR26	612				External S-RAM / Flash address	
EEA_17	AP26	729				External S-RAM / Flash address	
EEA_16	AW31	69				External S-RAM / Flash address	
EEA_15	AU29	357	O	U1	VDD3	External S-RAM / Flash address	
EEA_14	AU30	358				External S-RAM / Flash address	
EEA_13	AT29	490				External S-RAM / Flash address	
EEA_12	AT28	489				External S-RAM / Flash address	
EEA_11	AW33	71				External S-RAM / Flash address	
EEA_10	AR28	614				External S-RAM / Flash address	
EEA_9	AP28	731				External S-RAM / Flash address	
EEA_8	AU32	360				External S-RAM / Flash address	
EEA_7	AT30	491	O	U1	VDD3	External S-RAM / Flash address	
EEA_6	AT31	492				External S-RAM / Flash address	
EEA_5	AW35	73				External S-RAM / Flash address	
EEA_4	AW36	74				External S-RAM / Flash address	
EEA_3	AV35	222				External S-RAM / Flash address	
EEA_2	AU34	362				External S-RAM / Flash address	
EEA_1	AT32	493				External S-RAM / Flash address	
EEA_0	AR31	617				External S-RAM / Flash address	
EED_63	AT7	468	I/O	X	VDD3	External S-RAM / Flash data	
EED_62	AR9	595				External S-RAM / Flash data	
EED_61	AP10	713				External S-RAM / Flash data	
EED_60	AV5	192				External S-RAM / Flash data	
EED_59	AW4	42				External S-RAM / Flash data	
EED_58	AU7	335				External S-RAM / Flash data	
EED_57	AV6	193				External S-RAM / Flash data	
EED_56	AW5	43				External S-RAM / Flash data	

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
EED_55	AT9	470	I/O	X	VDD3	External S-RAM / Flash data	
EED_54	AV7	194				External S-RAM / Flash data	
EED_53	AW6	44				External S-RAM / Flash data	
EED_52	AU9	337				External S-RAM / Flash data	
EED_51	AR12	598				External S-RAM / Flash data	
EED_50	AW7	45				External S-RAM / Flash data	
EED_49	AU10	338				External S-RAM / Flash data	
EED_48	AV9	196				External S-RAM / Flash data	
EED_47	AU11	339	I/O	X	VDD3	External S-RAM / Flash data	
EED_46	AV10	197				External S-RAM / Flash data	
EED_45	AT13	474				External S-RAM / Flash data	
EED_44	AR14	600				External S-RAM / Flash data	
EED_43	AP14	717				External S-RAM / Flash data	
EED_42	AT14	475				External S-RAM / Flash data	
EED_41	AV11	198				External S-RAM / Flash data	
EED_40	AV12	199				External S-RAM / Flash data	
EED_39	AR15	601	I/O	X	VDD3	External S-RAM / Flash data	
EED_38	AV14	201				External S-RAM / Flash data	
EED_37	AV13	200				External S-RAM / Flash data	
EED_36	AR16	602				External S-RAM / Flash data	
EED_35	AW12	50				External S-RAM / Flash data	
EED_34	AT16	477				External S-RAM / Flash data	
EED_33	AU16	344				External S-RAM / Flash data	
EED_32	AW14	52				External S-RAM / Flash data	
EED_31	AU17	345	I/O	X	VDD3	External S-RAM / Flash data	
EED_30	AV15	202				External S-RAM / Flash data	
EED_29	AU18	346				External S-RAM / Flash data	
EED_28	AV16	203				External S-RAM / Flash data	
EED_27	AP18	721				External S-RAM / Flash data	
EED_26	AT19	480				External S-RAM / Flash data	
EED_25	AV17	204				External S-RAM / Flash data	
EED_24	AV18	205				External S-RAM / Flash data	

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
EED_23	AW18	56	I/O	X	VDD3	External S-RAM / Flash data	
EED_22	AT20	481				External S-RAM / Flash data	
EED_21	AW20	58				External S-RAM / Flash data	
EED_20	AP20	723				External S-RAM / Flash data	
EED_19	AW22	60				External S-RAM / Flash data	
EED_18	AV22	209				External S-RAM / Flash data	
EED_17	AW23	61				External S-RAM / Flash data	
EED_16	AT21	482				External S-RAM / Flash data	
EED_15	AW24	62	I/O	X	VDD3	External S-RAM / Flash data	
EED_14	AU22	350				External S-RAM / Flash data	
EED_13	AP22	725				External S-RAM / Flash data	
EED_12	AV25	212				External S-RAM / Flash data	
EED_11	AU23	351				External S-RAM / Flash data	
EED_10	AV24	211				External S-RAM / Flash data	
EED_9	AT23	484				External S-RAM / Flash data	
EED_8	AW26	64				External S-RAM / Flash data	
EED_7	AW27	65	I/O	X	VDD3	External S-RAM / Flash data	
EED_6	AW28	66				External S-RAM / Flash data	
EED_5	AV26	213				External S-RAM / Flash data	
EED_4	AV27	214				External S-RAM / Flash data	
EED_3	AU26	354				External S-RAM / Flash data	
EED_2	AU25	353				External S-RAM / Flash data	
EED_1	AW29	67				External S-RAM / Flash data	
EED_0	AR25	611				External S-RAM / Flash data	
MDD_31	J38	254	O	U1	VDD3	Real time monitor data	
MDD_30	H39	108				Real time monitor data	
MDD_29	M38	251				Real time monitor data	
MDD_28	R37	387				Real time monitor data	
MDD_27	P37	388				Real time monitor data	
MDD_26	P38	249				Real time monitor data	
MDD_25	U35	639				Real time monitor data	
MDD_24	R39	101				Real time monitor data	

MB91460 Series

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
MDD_23	V36	515	O	U1	VDD3	Real time monitor data	
MDD_22	V37	384				Real time monitor data	
MDD_21	W37	383				Real time monitor data	
MDD_20	AA39	95				Real time monitor data	
MDD_19	AB39	94				Real time monitor data	
MDD_18	AA35	635				Real time monitor data	
MDD_17	AA37	381				Real time monitor data	
MDD_16	AB34	749				Real time monitor data	
MDD_15	AB36	511	O	U1	VDD3	Real time monitor data	
MDD_14	AB37	380				Real time monitor data	
MDD_13	AC36	510				Real time monitor data	
MDD_12	AD34	747				Real time monitor data	
MDD_11	AE37	377				Real time monitor data	
MDD_10	AF38	237				Real time monitor data	
MDD_9	AE35	631				Real time monitor data	
MDD_8	AJ38	234				Real time monitor data	
MDD_7	AF34	745	O	U1	VDD3	Real time monitor data	
MDD_6	AK38	233				Real time monitor data	
MDD_5	AJ37	373				Real time monitor data	
MDD_4	AJ36	504				Real time monitor data	
MDD_3	AH34	743				Real time monitor data	
MDD_2	AK36	503				Real time monitor data	
MDD_1	AK35	626				Real time monitor data	
MDD_0	AM36	501				Real time monitor data	
MDDEN_3	M34	759	O	U1	VDD3	Real time monitor control	
MDDEN_2	H38	255				Real time monitor control	
MDDEN_1	L36	522				Real time monitor control	
MDDEN_0	L37	391				Real time monitor control	
RCLK	H37	394	O	U1	VDD3	Real time monitor control	
MRSTX	J36	524				Real time monitor control	
DHIT_5	H34	763	O	U1	VDD3	Real time monitor data hit	
DHIT_4	G35	649				Real time monitor data hit	
DHIT_3	H36	525				Real time monitor data hit	

Pin Name	JEDEC	Pin no.	I/O	I/O Circ. type *1	Power do-main *2	Pin function	Notes
DHIT_2	G37	395	O	U1	VDD3	Real time monitor data hit	
DHIT_1	F38	257				Real time monitor data hit	
DHIT_0	E39	111				Real time monitor data hit	
IHIT_3	M3	306	O	U1	VDD3	Real time monitor instruction hit	
IHIT_2	P5	570				Real time monitor instruction hit	
IHIT_1	N4	442				Real time monitor instruction hit	
IHIT_0	K2	161				Real time monitor instruction hit	
FLASH_RDY	G1	7	I	T1	VDD3	External flash memory control	
FLASH_SYNC	M5	568	O	U1	VDD3	External flash memory control	
FLASH_ATDIN	M4	441				External flash memory control	
FLASH_EQIN	K3	304				External flash memory control	
FLASH_RD32	H2	159				External flash memory control	
FLASH_BYTEX	H3	302				External flash memory control	
FLASH_SECUR	F1	6				External flash memory control	
FLASH_FRSTX	G3	301				External flash memory control	
CSV_MM	A5	149	O	M	VDD5	Clock Supervisor: Main clock missing	
CSV_SM	A4	150				Clock Supervisor: Sub clock missing	
REGC_3	E6	680	O	P	VDD5	External voltage regulator control	N-channel open drain output
REGC_2	D5	560				External voltage regulator control	
REGC_1	D6	559				External voltage regulator control	
REGC_0	A3	151				External voltage regulator control	
REGS	C4	432	I	H2	VDD5	External voltage regulator status	
AVSS0	B4	295	I	Z0	VDD5	Analog ground for ADC0, DAC, ALARM	
AVSS1	AH4	457				Analog ground for ADC1	
AVRL0	E8	678				Analog low reference for ADC0	
AVRL1	AN1	33				Analog low reference for ADC1	
AVRH50	D7	558	I	Z1	VDD5	Analog high reference for ADC0	
AVRH51	AN2	184				Analog high reference for ADC1	
AVCC50	F10	789				Analog power for ADC0, DAC, ALARM	
AVCC51	AM2	183				Analog power for ADC1	

1. About the types of I/O cells, please refer to chapter “I/O Circuit Types MB91FV460B” on page 179.
2. About the I/O power domains, please refer to section “Power Supply Pins” on page 175.

MB91460 Series

3.8.9 Power Supply Pins

The following tables list the power supply pins of MB91FV460B:

Table 3.8-8 Common ground pin numbers

VSS	Common ground					pin numbers	
1	2	38	39	40	76	77	78
114	115	116	152	153	189	225	261
793	794	796	798	800	802	804	805
806	807	809	811	813	815	817	818
819	820	822	824	826	828	830	831
832	833	835	837	839	841	843	844
845	846	848	850	852	854	856	857
858	859	861	863	865	867	869	870
871	872	874	876	878	880	882	883
884	885	887	889	891	893	895	896

Table 3.8-9 Common ground JEDEC numbers

VSS	Common ground					JEDEC	
A1	B1	AV1	AW1	AW2	AW38	AW39	AV39
B39	A39	A38	A2	B2	AV2	AV38	B38
G7	H7	K7	M7	P7	T7	V7	W7
Y7	AA7	AC7	AE7	AG7	AJ7	AL7	AM7
AN7	AN8	AN10	AN12	AN14	AN16	AN18	AN19
AN20	AN21	AN23	AN25	AN27	AN29	AN31	AN32
AN33	AM33	AK33	AH33	AF33	AD33	AB33	AA33
Y33	W33	U33	R33	N33	L33	J33	H33
G33	G32	G30	G28	G26	G24	G22	G21
G20	G19	G17	G15	G13	G11	G9	G7

Table 3.8-10 Core supply 1.8V pin numbers

VDDI	Core power supply 1.8V					pin numbers	
682	684	686	688	690	692	694	696
698	700	702	704	706	708	710	712
714	716	718	720	722	724	726	728
730	732	734	736	738	740	742	744
746	748	750	752	754	756	758	760
762	764	766	768	770	772	774	776
778	780	782	784	786	788	790	792

Table 3.8-11 Core supply 1.8V JEDEC numbers

VDDI	Core power supply 1.8V					JEDEC	
G6	J6	L6	N6	R6	U6	W6	AA6
AC6	AE6	AG6	AJ6	AL6	AN6	AP7	AP9
AP11	AP13	AP15	AP17	AP19	AP21	AP23	AP25
AP27	AP29	AP31	AP33	AN34	AL34	AJ34	AG34
AE34	AC34	AA34	W34	U34	R34	N34	L34
J34	G34	F33	F31	F29	F27	F25	F23
F21	F19	F17	F15	F13	F11	F9	F7

Table 3.8-12 IO ring power pin numbers

VDD5	IO ring power 5V					pin numbers	
11	49	167	331	333	435	454	459
567	579	599	873	875	877	879	881
886	888	890	892	894			

Table 3.8-13 IO ring power JEDEC numbers

VDD5	IO ring power 5V					JEDEC	
L1	AW11	T2	AU3	AU5	F4	AE4	AK4
L5	AC5	AR13	G31	G29	G27	G25	G23
G18	G16	G14	G12	G10			

Table 3.8-14 DSU4 IO power 3.3V pin numbers

VDD3	DSU4 IO power 3.3V					pin numbers	
83	97	102	110	113	376	379	389
621	795	797	799	801	803	808	810
812	814	816	821	823	825	827	829
834	836	838	840	842			

Table 3.8-15 DSU4 IO power 3.3V JEDEC numbers

VDD3	DSU4 IO power 3.3V					JEDEC	
AN39	W39	P39	F39	C39	AF37	AC37	N37
AR35	J7	L7	N7	R7	U7	AB7	AD7
AF7	AH7	AK7	AN9	AN11	AN13	AN15	AN17
AN22	AN24	AN26	AN28	AN30			

MB91460 Series**Table 3.8-16 External bus power 3V to 5V pin numbers**

VDD35	External bus power 3V to 5V					pin numbers	
847	849	851	853	855	860	862	864
866	868						

Table 3.8-17 External bus power 3V to 5V JEDEC numbers

VDD35	External bus power 3V to 5V					JEDEC	
AL33	AJ33	AG33	AE33	AC33	V33	T33	P33
M33	K33						

Table 3.8-18 USB power 3.3V pin numbers

VDDU	USB power 3.3V					pin numbers	
75							

Table 3.8-19 USB power 3.3V JEDEC numbers

VDDU	USB power 3.3V					JEDEC	
AW37							

Table 3.8-20 Media LB IO power 3.3V pin numbers

VDDM	Media LB IO power 3.3V					pin numbers	
54	59	216	218	219	607		

Table 3.8-21 Media LB IO power 3.3V JEDEC numbers

VDDM	Media LB IO power 3.3V					JEDEC	
AW16	AW21	AV29	AV31	AV32	AR21		

Table 3.8-22 Stepper motor power 3V to 5V pin numbers

HVDD	Stepper motor power 3V to 5V					pin numbers	
132	137	140	409	543	668		

Table 3.8-23 Stepper motor power 3V to 5V JEDEC numbers

HVDD	Stepper motor power 3V to 5V					JEDEC	
A22	A17	A14	C27	D22	E18		

Table 3.8-24 APIX supply pin numbers

VDDA	APIX VDD					pin numbers	
603							

Table 3.8-25 APIX supply JEDEC numbers

VDDA	APIX VDD					JEDEC	
AR17							

Table 3.8-26 APIX ground pin numbers

VSSA	APIX ground					pin numbers	
51	347						

Table 3.8-27 APIX ground JEDEC numbers

VSSA	APIX ground					JEDEC	
AW13	AU19						

MB91460 Series**3.9. I/O Circuit Types MB91FV460B****3.9.1 I/O Circuit Overview**

The following table lists the I/O circuit properties of MB91FV460B.

Table 3.9-1 MB91FV460B I/O circuit type overview

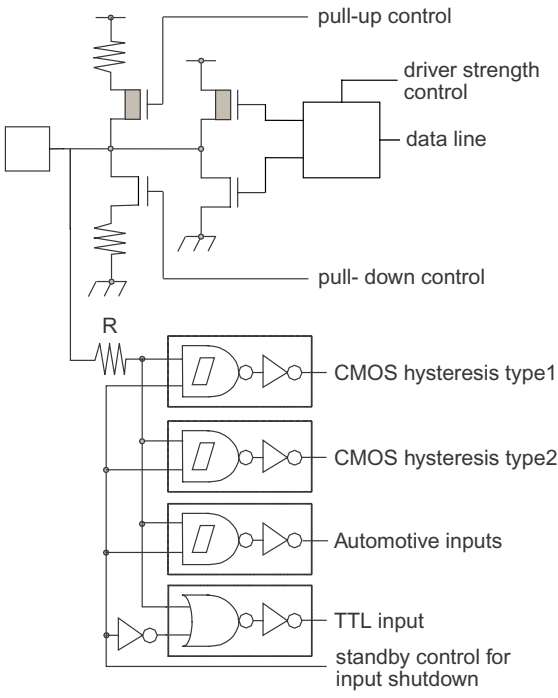
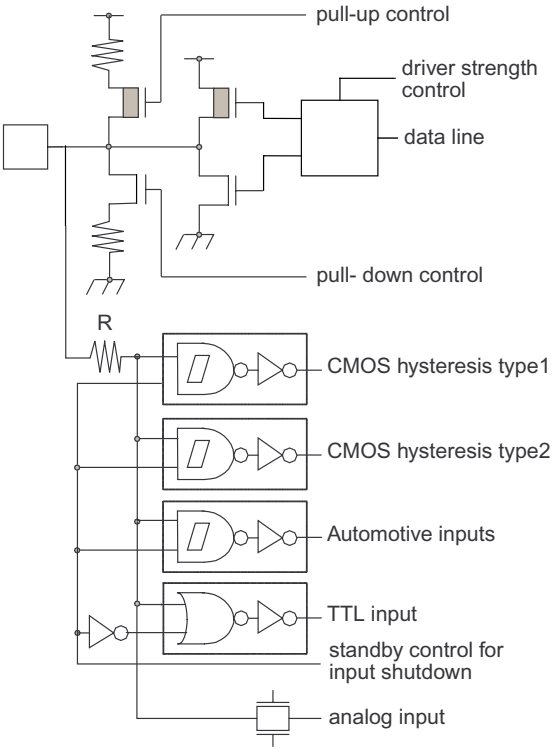
I/O Type	I/O	Input Level	Pull-Up/Down	Output Driver	Analog or Special Function	Comments
A	I/O	CS/A/T/CS2 * ¹	Up/Dn 50 k Ω * ²	2/5 mA * ³	-	
B1	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA	1 analog input	ADC input
B2	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA	1 analog input 1 analog output	ADC input DAC output
C	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	3 mA	-	I2C I/O cell
D	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	3 mA	1 analog input	I2C I/O cell ADC input
E	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA 30 mA	-	SMC I/O cell
F	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA 30 mA	1 analog input	SMC I/O cell ADC input
G	I	CMOS Schmitt	-	-	12 V withstand (for MD [2:0])	
H1	I	CMOS Schmitt	Pull-Up 50 k Ω	-	-	
H2	I	CMOS Schmitt	-	-	-	
H3	I	CMOS Schmitt	Pull-Down 50 k Ω	-	-	
J1	I/O	4 MHz / High-speed Oscillator				
J2	I/O	32 kHz Oscillator				
K	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA	1 SEG/COM out	LCD Segment/COM
K1	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA	1 SEG/COM out, 1 analog input	LCD Segment/COM ADC input
L	I/O	CS/A/T/CS2	Up/Dn 50 k Ω	2/5 mA	1 VLCD input	LCD voltage input
M	O	-	-	5 mA	CMOS tri-state output	
N1	I	analog	-	-	Analog input pin with protection diodes	
N2	O	-	-	analog	Analog output pin with protection diodes	
O	O	-	-	5 mA	CMOS output	
P	O	-	-	5 mA	N-Channel open drain output	
Q	I/O	USB	-	USB	USB differential I/O	
R	I/O	C/CS * ⁴	Up/Dn 33 k Ω	4 mA	-	for 3V I/O
S	I/O	C/CS/MLB * ⁵	Up/Dn 33 k Ω	6 mA	-	Media LB I/O, 3V
T1	I	CMOS	-	-	-	
T2	I	CMOS	Down 33 k Ω ctrl.	-	-	Pull-down controllable
T3	I	CMOS Schmitt	Down 33 k Ω ctrl.	-	-	Pull-down controllable
T4	I	CMOS Schmitt	Down 33 k Ω	-	-	

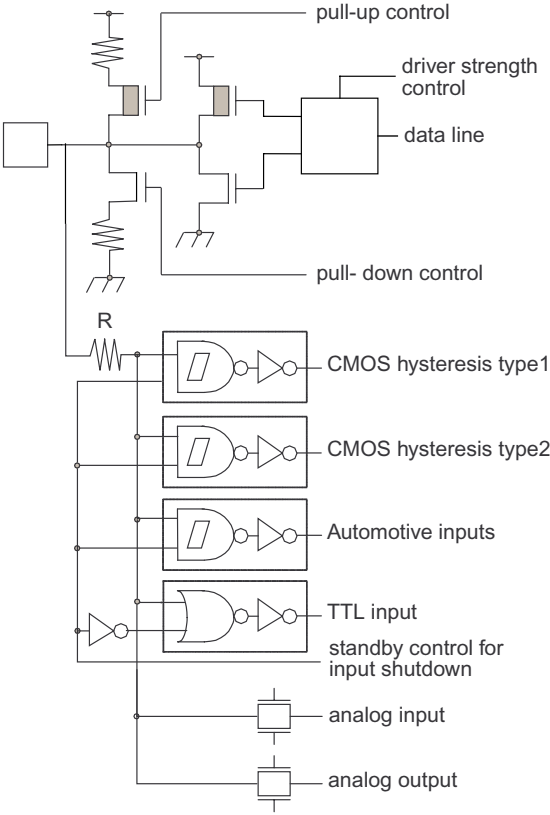
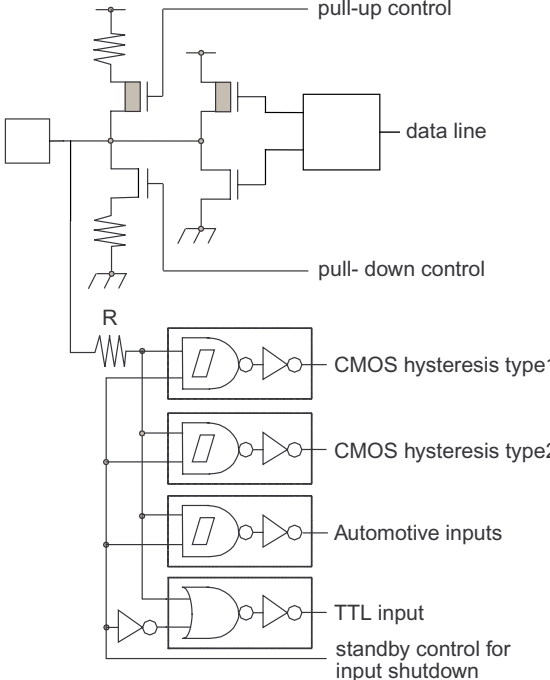
I/O Type	I/O	Input Level	Pull-Up/Down	Output Driver	Analog or Special Function	Comments
TR	I	CMOS Schmitt	Up 33 kΩ	-		special input for DSU4 reset (TRSTX)
U0	O	-	-	4 mA	-	
U1	O	-	-	4 mA		initial L output level
U2	O	-	-	4 mA		initial H output level
V	O	-	-	8 mA	-	
W	I/O	CMOS Schmitt	-	10 mA	-	
X	I/O	CMOS	-	4 mA	-	
Z0	-	-	-	-		analog ground supply with protection diodes
Z1	-	-	-	-		analog power supply with protection diodes

- Selectable by control register (PILR, EPILR):
 - CMOS-Schmitt 0.3/0.7 Vdd
 - Automotive 0.5/0.8 Vdd
 - TTL 0.8V / 2.0V
 - CMOS-Schmitt 2 0.2/0.8 Vdd
- Selectable by control register: Disabled or Pull-up or Pull-down
- Selectable driver strength by control register (PODR)
- Selectable by control register (PILR):
 - CMOS 0.3/0.7 Vdd
 - CMOS-Schmitt 0.2/0.8 Vdd
- Selectable by control register (PILR, EPILR):
 - CMOS 0.3/0.7 Vdd
 - CMOS-Schmitt 0.2/0.8 Vdd
 - Media LB 0.7V / 1.7V

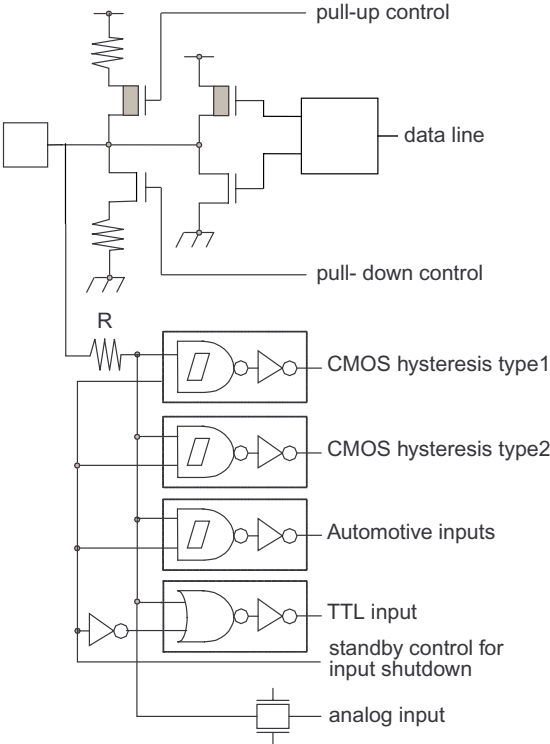
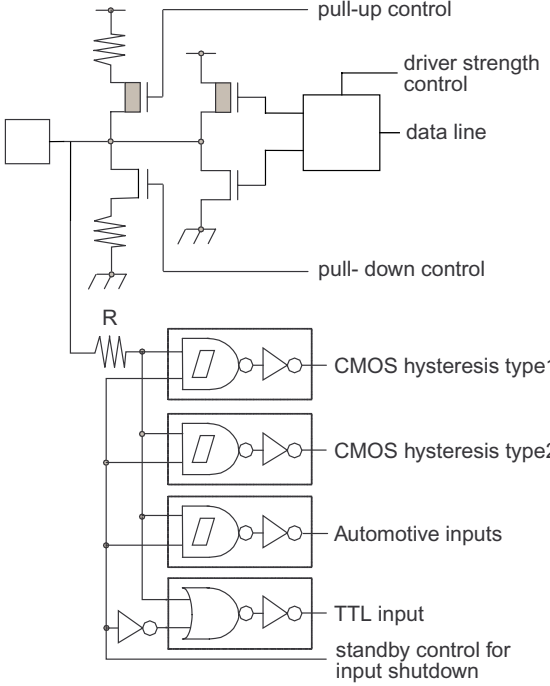
MB91460 Series

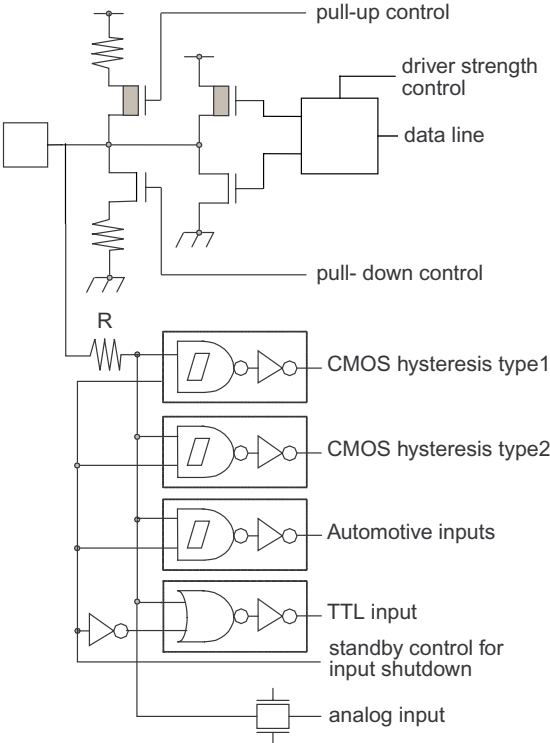
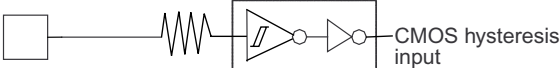
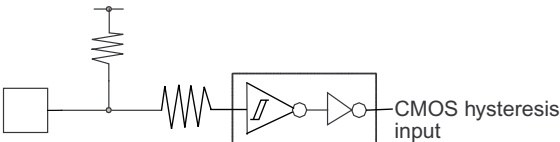
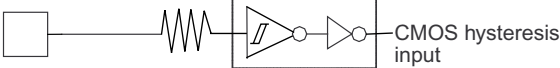
3.9.2 I/O Circuit Description

Type	Circuit	Remarks
A		<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p>
B1		<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>Analog input</p>

Type	Circuit	Remarks
B2	 <p>The diagram for Type B2 shows a complex I/O circuit. It includes a pull-up control and a pull-down control connected to a data line. A driver strength control is also present. The circuit features two CMOS hysteresis inputs (type 1 and type 2), Automotive inputs, and a TTL input with a standby control for input shutdown. Additionally, there are analog input and analog output sections. A resistor R is connected to the input lines.</p>	<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>Analog input Analog output</p>
C	 <p>The diagram for Type C shows a similar I/O circuit to Type B2. It includes a pull-up control and a pull-down control connected to a data line. The circuit features two CMOS hysteresis inputs (type 1 and type 2), Automotive inputs, and a TTL input with a standby control for input shutdown. Additionally, there are analog input and analog output sections. A resistor R is connected to the input lines.</p>	<p>CMOS level output ($I_{OL} = 3\text{mA}$, $I_{OH} = -3\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p>

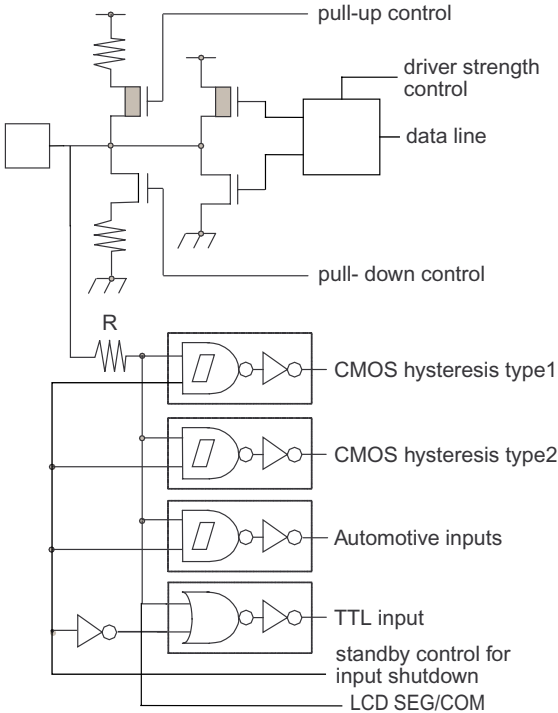
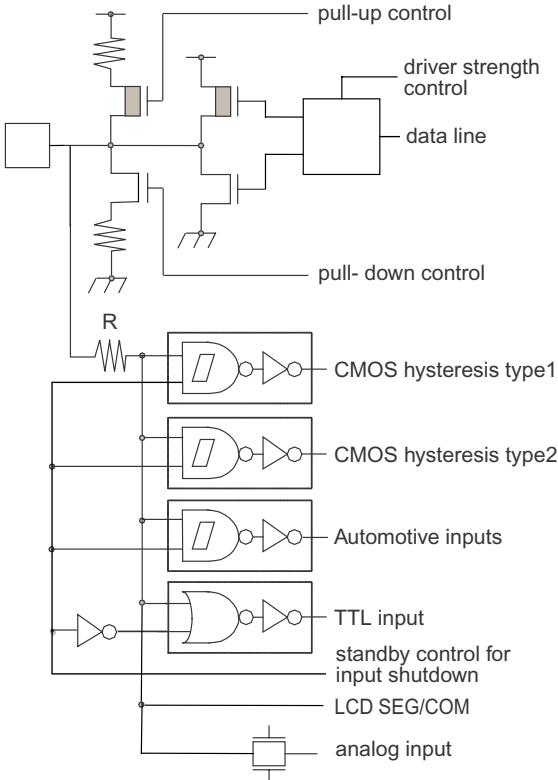
MB91460 Series

Type	Circuit	Remarks
D		CMOS level output ($I_{OL} = 3\text{mA}$, $I_{OH} = -3\text{mA}$) 2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function Programmable pull-up resistor: 50k Ω approx. Programmable pull-down resistor: 50k Ω appr. Analog input
E		CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$, and $I_{OL} = 30\text{mA}$, $I_{OH} = -30\text{mA}$) 2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function Programmable pull-up resistor: 50k Ω approx. Programmable pull-down resistor: 50k Ω appr.

Type	Circuit	Remarks
F		<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$, and $I_{OL} = 30\text{mA}$, $I_{OH} = -30\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>Analog input</p>
G		<p>Mask ROM device: CMOS Hysteresis input pin Flash device: CMOS input pin 12 V withstand (for MD [2:0])</p>
H1		<p>CMOS Hysteresis input pin</p> <p>Pull-up resistor value: 50 kΩ approx.</p>
H2		<p>CMOS Hysteresis input pin</p>

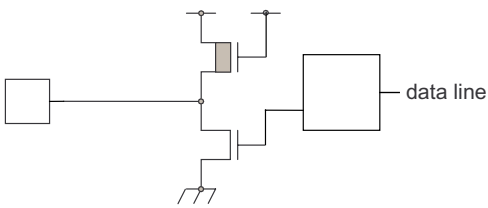
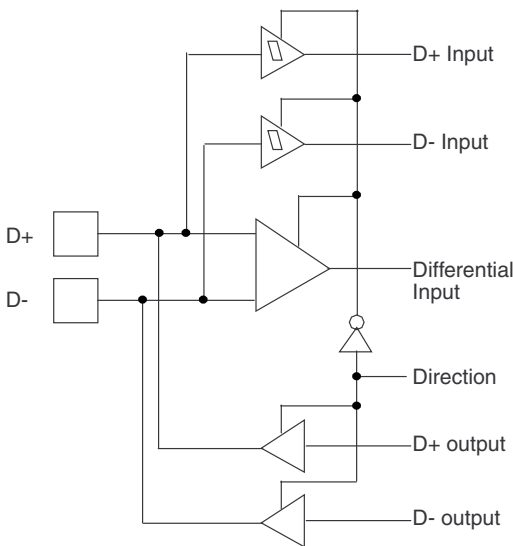
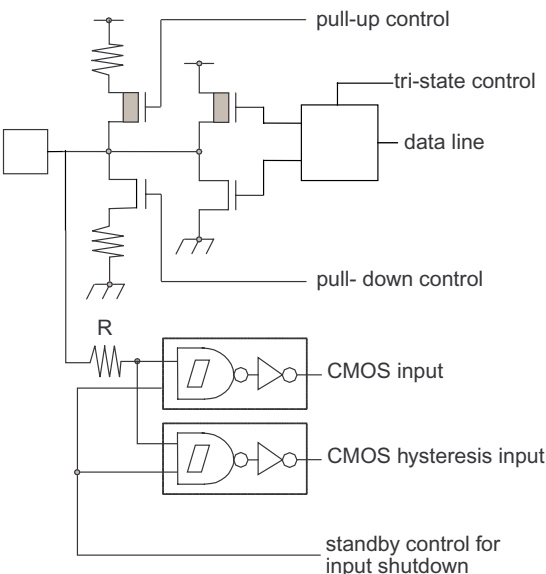
MB91460 Series

Type	Circuit	Remarks
H3		CMOS Hysteresis input pin Pull-down resistor value: 50 kΩ approx.
J1		High-speed oscillation circuit: <ul style="list-style-type: none">• Programmable between oscillation mode (external crystal or resonator connected to X0/X1 pins) and Fast external Clock Input (FCI) mode (external clock connected to X0 pin)• Feedback resistor = approx. 2 * 0.5 MΩ. Feedback resistor is grounded in the center when the oscillator is disabled or in FCI mode.
J2		Low-speed oscillation circuit: <ul style="list-style-type: none">• Programmable between oscillation mode (external crystal or resonator connected to X0/X1 pins) and Fast external Clock Input (FCI) mode (external clock connected to X0A pin)• Feedback resistor = approx. 20 MΩ. (X1A side: 19.5 MΩ, X0A side: 0.5 MΩ) Feedback resistors are grounded in the center when the oscillator is disabled.

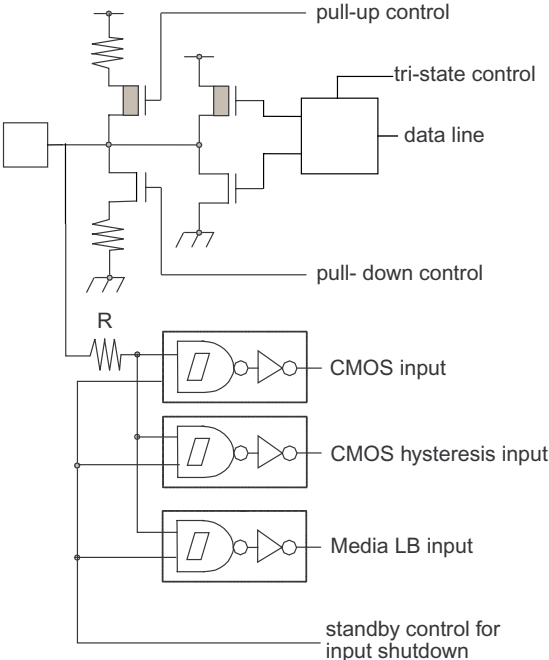
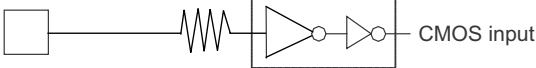
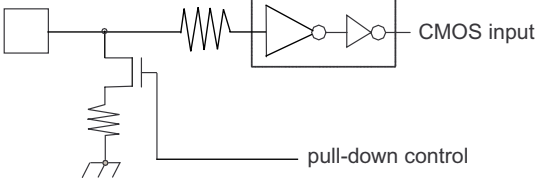
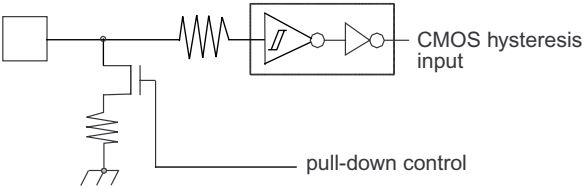
Type	Circuit	Remarks
K	 <p>The diagram for Type K shows a data line connected to a driver with pull-up and pull-down controls. A resistor R is connected to the data line. Below the data line, there are four input types: CMOS hysteresis type1, CMOS hysteresis type2, Automotive inputs, and TTL input. A standby control for input shutdown is also shown, along with an LCD SEG/COM output.</p>	<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>LCD SEG/COM output</p>
K1	 <p>The diagram for Type K1 is similar to Type K but includes an additional analog input at the bottom. The data line, driver controls, resistor R, and input types (CMOS hysteresis types 1 and 2, Automotive inputs, TTL input, standby control for input shutdown, and LCD SEG/COM output) are the same as in Type K.</p>	<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>LCD SEG/COM output Analog input</p>

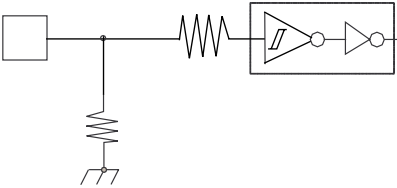
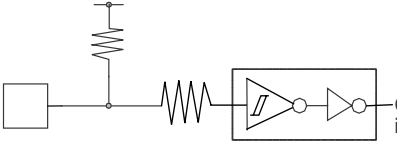
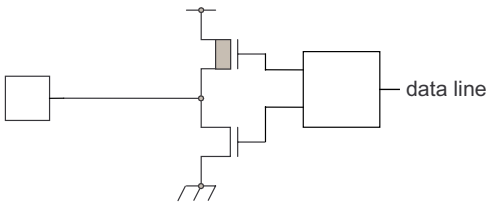
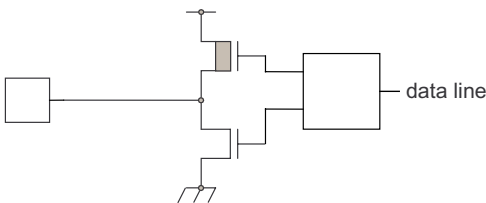
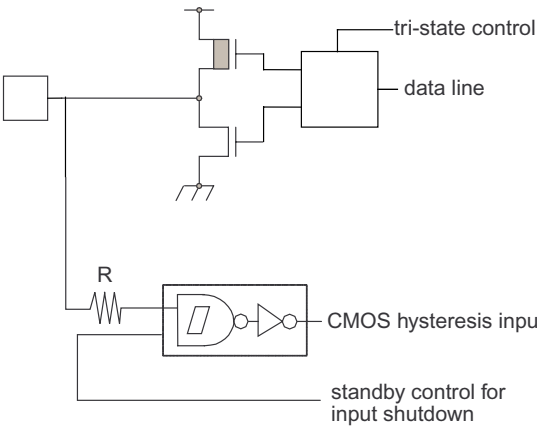
MB91460 Series

Type	Circuit	Remarks
L		<p>CMOS level output (programmable $I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$ and $I_{OL} = 2\text{mA}$, $I_{OH} = -2\text{mA}$)</p> <p>2 different CMOS hysteresis inputs with input shutdown function Automotive input with input shutdown function) TTL input with input shutdown function</p> <p>Programmable pull-up resistor: 50kΩ approx. Programmable pull-down resistor: 50kΩ appr.</p> <p>Analog input LCD Voltage input</p>
M		<p>CMOS level tri-state output ($I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$)</p>
N1 N2		<p>Analog terminal</p> <p>Type N1: Analog input pin with protection Type N2: Analog output pin with protection</p>
O		<p>CMOS level output ($I_{OL} = 5\text{mA}$, $I_{OH} = -5\text{mA}$)</p>

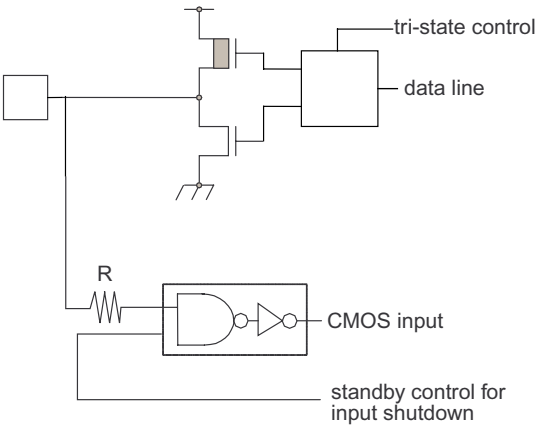
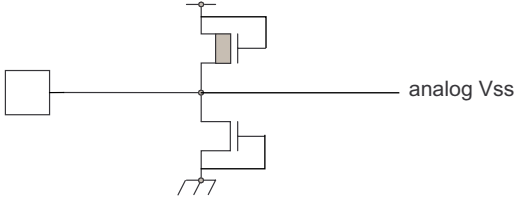
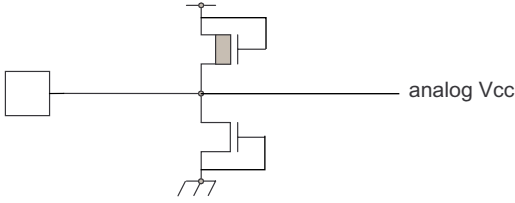
Type	Circuit	Remarks
P		N-Channel open drain output ($I_{OL} = 5\text{mA}$)
Q		USB IO cell UDP and UDM differential I/O
R		CMOS level output $I_{OL} = 4\text{mA}$, $I_{OH} = -4\text{mA}$ CMOS level input with shutdown function CMOS hysteresis input with shutdown function, Programmable pull-up resistor: $33\text{k}\Omega$ approx., Programmable pull-down resistor: $33\text{k}\Omega$ appr.

MB91460 Series

Type	Circuit	Remarks
S		CMOS level output $I_{OL} = 6\text{mA}$, $I_{OH} = -6\text{mA}$ CMOS input with shutdown function CMOS hysteresis input with shutdown function Media LB input with shutdown function Programmable pull-up resistor: 33k Ω approx., Programmable pull-down resistor: 33k Ω appr.
T1		CMOS input
T2		CMOS input Programmable pull-down resistor: 33k Ω appr.
T3		CMOS hysteresis input Programmable pull-down resistor: 33k Ω appr.

Type	Circuit	Remarks
T4		CMOS hysteresis input Pull-down resistor: 33kΩ appr.
TR		CMOS hysteresis input Fixed pull-up resistor: 33kΩ approx., Special input cell for TRSTX
U0 U1 U2		CMOS level output $I_{OL} = 4\text{mA}$, $I_{OH} = -4\text{mA}$ Type U1 has initial output L level. Type U2 has initial output H level.
V		CMOS level output $I_{OL} = 8\text{mA}$, $I_{OH} = -8\text{mA}$
W		CMOS level output $I_{OL} = 10\text{mA}$, $I_{OH} = -10\text{mA}$ CMOS hysteresis input with shutdown function

MB91460 Series

Type	Circuit	Remarks
X		CMOS level output $I_{OL} = 4\text{mA}$, $I_{OH} = -4\text{mA}$ CMOS input with shutdown function
Z0		Analog ground supply with protection device
Z1		Analog power supply with protection device

3.10. Pin State Table

The pin state table explains the state of the pins depending on the state and mode of the MB91460 series devices.

3.10.1 Explanation of the meaning of words and phrases

Word / phrase	Explanation
Input ON	It is possible to input a signal to the device.
Input GPORTEN	It is possible to input a signal to the device after control bit GPORTEN (Global Port Enable) in PORTEN register is set. GPORTEN is cleared by INIT state.
Input CPORTEN	It is possible to input a signal to the device after control bit CPORTEN (Communication Port Enable) in PORTEN register is set. CPORTEN is used by the Boot ROM to enable serial communication with the Softune system via USART4 and is cleared by INIT state.
Input bus hold	To prevent leakage by floating inputs, the input level is fixed internally to the value which was on the input immediately before changing the state (internal bus holder function).
Input fixed	To prevent leakage by floating inputs, the input level is fixed to "0" internally.
Output ON	The device always outputs data.
Output enabled	It is possible to output data from the device.
Output Hi-Z	The output is in Hi-Z state.
State hold	The state (input/output) of the pin immediately before changing the state is maintained. In case of output, the output value of the pin is maintained.
Pull-Up/Down enabled	The internal Pull-Up/Down is active (if enabled by control register setting).
Pull-Up/Down ON	The internal Pull-Up/Down is always ON.
Pull-Up/Down OFF	The internal Pull-Up/Down is switched off by hardware.

3.10.2 Explanation of the device states

State	Explanation
INIT/RST	Settings initialization reset (by pin INITX=0, hard-/software watchdog, or clock supervisor). The INIT state always includes a RST state.
RST	Operation initialization reset (software reset)
RUN	After reset and vector fetch, software is running.
SLEEP	SLEEP state (clocks ON, resources ON, CPU OFF).
STOP	STOP state (clocks OFF, resources OFF, CPU OFF) with HIZ = 0.
STOP-HIZ	STOP state (clocks OFF, resources OFF, CPU OFF) with HIZ = 1.

MB91460 Series

3.10.3 Initial Behaviour of the External Bus

The initial behaviour of the external bus differs from device to device and depending on the device mode (set by the pins MD_2, MD_1, MD_0):

Table 3.10-1 Initial behaviour of external bus

Device mode	Devices	External Bus Initial Behaviour
MD_2, MD_1, MD_0 = 000 ¹	MB91F464Ax MB91F465Kx MB91F465Cx MB91F465Dx MB91F467Cx MB91F467Dx MB91F469Gx	The external bus is enabled, the port function registers of the external bus ports are set by RST state. See pin state table 3.10.4 External bus pins in external bus mode (Page No.194) .
	MB91F465Px MB91F467Ex MB91F467Px MB91F467Sx MB91F467Tx MB91F469Tx MB91F469Qx	The external bus is disabled, the port function registers of the external bus ports are cleared by RST state. See pin state table 3.10.5 General purpose / resource pins and external bus pins in single chip mode (Page No.196) .
MD_2, MD_1, MD_0 = 001 ²	all devices	The external bus is enabled, see pin state table 3.10.4 External bus pins in external bus mode (Page No.194) .

1. Internal vector fetch mode
2. External vector fetch mode

On MB91FV460B, both types of external bus behaviour can be emulated by setting the appropriate series mode on the pins SER_MD2, SER_MD1, SER_MD0. See [3.8.1 MB91460 Series Modes \(Page No.143\)](#).

3.10.4 External bus pins in external bus mode

Table 3.10-2 Pin states of external bus pins in external bus mode

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
External bus data I/O ports (P00 - P03)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	ON	enabled	enabled	
	SLEEP	state hold	state hold	enabled	
	STOP	bus hold	state hold	enabled	
	STOP-HIZ	bus hold	Hi-Z	OFF	
External bus address output ports (P04 - P07)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	state hold	OFF	
	RUN	GPORTEN ¹	ON	OFF	Pull-Up/Down OFF because the pins are always in output direction. On MB91F467Dx, the Pull-Up/Downs are enabled.
	SLEEP	state hold	state hold	OFF	
	STOP	bus hold	state hold	OFF	
	STOP-HIZ	bus hold	Hi-Z	OFF	
External bus control inputs: RDY (P08_7) BRQ (P08_6) MCLKI (P10_5)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	GPORTEN ²	Hi-Z	enabled	MCLKI input is ON
	SLEEP	state hold	state hold	enabled	
	STOP	bus hold	state hold	enabled	
	STOP-HIZ	bus hold	Hi-Z	OFF	
Ext. bus control outputs: BGRNTX (P08_5) RDX (P08_4) WRX3:0 (P08_3:0) CSX7:0 (P09_7:0) MCLKE (P10_6) WEX (P10_3) BAAX (P10_2) ASX (P10_1)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	state hold	OFF	
	RUN	GPORTEN ¹	ON	OFF	Pull-Up/Down OFF because the pins are always in output direction. On MB91F467Dx, the Pull-Up/Downs are enabled.
	SLEEP	state hold	state hold	OFF	
	STOP	bus hold	state hold	OFF	
	STOP-HIZ	bus hold	Hi-Z	OFF	
External bus clock outputs: MCLKO (P10_4) SYSCLK (P10_0)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	state hold	OFF	
	RUN	GPORTEN ¹	ON	OFF	Pull-Up/Down OFF because the pins are always in output direction. On MB91F467Dx, the Pull-Up/Downs are enabled.
	SLEEP	state hold	state hold	OFF	
	STOP	bus hold	state hold	OFF	
	STOP-HIZ	bus hold	Hi-Z	OFF	

MB91460 Series

3.10.Pin State Table

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
DMA control inputs: DREQ3/2 (P12_4/0) DREQ1/0 (P13_4/0) DEOTX3/2 (P12_6/2) DEOTX1/0 (P13_6/2)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	GPORTEN	enabled	enabled	
	SLEEP	state hold	state hold	enabled	
	STOP	bus hold	state hold	enabled	
	STOP-HIZ	bus hold	Hi-Z	OFF	
DMA control outputs: DACKX3/2 (P12_5/1) DACKX1/0 (P13_5/1) DEOP3/2 (P12_7/3) DEOP1/0 (P13_7/3) IOWRX (P11_1) IORDX (P11_0)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	GPORTEN ¹	enabled	OFF	Pull-Up/Down OFF because the pins are always in output direction. On MB91F467Dx, the Pull-Up/Downs are enabled.
	SLEEP	state hold	state hold	OFF	
	STOP	bus hold	state hold	OFF	
	STOP-HIZ	bus hold	Hi-Z	OFF	

1. GPORTEN=1 opens the input lines, but the ports work in output direction always.
2. GPORTEN=1 is needed to use RDY and BRQ features. MCLKI input is always ON.

3.10.5 General purpose / resource pins and external bus pins in single chip mode**Table 3.10-3 Pin states of general purpose ports and external bus pins in single chip mode**

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
Pins of the Debug USART (USART4): SOT4 (P19_1) SIN4 (P19_0)	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	CPORTEN GPORTEN ¹	enabled ²	enabled	
	SLEEP	state hold	state hold	enabled	
	STOP	bus hold	state hold	enabled	
	STOP-HIZ	bus hold	Hi-Z	OFF	
All pins of the ports P14 to P40, which have an external interrupt feature (INT0 to INT31), and the interrupt is enabled by PFR and ENIR setting	INIT/RST	bus hold	Hi-Z	OFF	External interrupts are not enabled in INIT/RST and RST state
	RST	state hold	Hi-Z	OFF	
	RUN	ON ³	enabled	enabled	
	SLEEP	ON	state hold	enabled	
	STOP	ON	state hold	enabled	
	STOP-HIZ	ON	Hi-Z	OFF	
All pins of the ports P14 to P40 which are not mentioned above and P00 to P13 in single chip mode	INIT/RST	bus hold	Hi-Z	OFF	
	RST	state hold	Hi-Z	OFF	
	RUN	GPORTEN ⁴	enabled	enabled	
	SLEEP	state hold	state hold	enabled	
	STOP	bus hold	state hold	enabled	
	STOP-HIZ	bus hold	Hi-Z	OFF	

1. During execution of Boot ROM, CPORTEN is set and SIN4 input is ON. After Boot ROM, SIN4 depends on CPORTEN or GPORTEN.
2. During execution of Boot ROM, SOUT4 output is ON.
3. External interrupt enable overwrites GPORTEN. The input line is ON also if GPORTEN=0. External interrupt enable is cleared by RST. GPORTEN is cleared by INIT.
4. Ports P00 to P03 and P10[5] (MCLK) do not depend on GPORTEN. Input is always enabled.

MB91460 Series

3.10.Pin State Table

3.10.6 Special I/O pins

Table 3.10-4 Pin states of special I/O pins

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
Main Oscillator X0, X1 Sub Oscillator X0A, X1A	INIT/RST	ON	ON	n.a.	Sub Oscillator is OFF.
	RST	ON	ON		
	RUN	ON	ON		
	SLEEP	ON	ON		
	STOP	ON/OFF	ON/OFF		Main osc. is OFF if OSCD1=1
	STOP-HIZ	ON/OFF	ON/OFF		Sub osc. is OFF if OSCD2=1
Device control inputs INITX RSTX ¹ HSTX NMIX	INIT/RST	ON	n.a.	Pull Up ON	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			
Device mode inputs ² MD_2 MD_1 MD_0	INIT/RST	ON	n.a.	n.a.	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			
Fast Clock Input control FCI ³	INIT/RST	ON	n.a.	Pull DownON	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			
Clock Monitor output MONCLK	INIT/RST	n.a.	Hi-Z	n.a.	output depends on CMCFG register content
	RST		Hi-Z		
	RUN		enabled		
	SLEEP		enabled		
	STOP		enabled		
	STOP-HIZ		Hi-Z		

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
Alarm Comparator inputs ALARM_1 ALARM_0	INIT/RST	ON	n.a.	n.a.	RST clears the PD (power down) control bit.
	RST	ON			Input is enabled as long as the Alarm comparator channel is enabled (PD=0).
	RUN	enabled			
	SLEEP	enabled			
	STOP	OFF			
	STOP-HIZ	OFF			
APIX SDOUT	INIT/RST	n.a.	OFF	n.a.	
	RST		OFF		
	RUN		enabled		
	SLEEP		enabled		
	STOP		enabled		
	STOP-HIZ		enabled		
APIX SDIN	INIT/RST	OFF	n.a.	n.a.	
	RST	OFF			
	RUN	enabled			
	SLEEP	enabled			
	STOP	enabled			
	STOP-HIZ	enabled			
USB UDM/UDP	INIT/RST	OFF	OFF	n.a.	
	RST	OFF	OFF		
	RUN	enabled	enabled		
	SLEEP	enabled	enabled		
	STOP	OFF	OFF		
	STOP-HIZ	OFF	OFF		
USB HCONX	INIT/RST	n.a.	output H	n.a.	
	RST		enabled		
	RUN		enabled		
	SLEEP		enabled		
	STOP		enabled		
	STOP-HIZ		enabled		

1. RSTX, HSTX, NMIX are not available on most devices
2. There may be a MD_3 input on some devices. Its function depends on the device.
3. FCI is not available on most devices. On some devices, the FCI pin is named MD_3.

MB91460 Series**3.10.7 Special I/O pins on MB91FV460B**

The following table is valid for MB91FV460B only.

Table 3.10-5 Pin states of MB91FV460B special I/O pins

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
Clock Supervisor outputs CSV_MM, CSV_SM	INIT/RST	n.a.	Hi-Z	n.a.	output depends on CSVCR_OUTE control bit
	RST		Hi-Z		
	RUN		enabled		
	SLEEP		enabled		
	STOP		enabled		
	STOP-HIZ		enabled		
Hardware watchdog reset output WDRESX	INIT/RST	n.a.	Hi-Z / enabled	n.a.	Hi-Z if external INITX is applied.
	RST		enabled		
	RUN		enabled		
	SLEEP		enabled		
	STOP		enabled		
	STOP-HIZ		enabled		
Device control inputs CSV_KILL EDSU_BREAKX FIX_ENX HWWDG_KILL SRAM_SFX	INIT/RST	ON	n.a.	Pull Up ON	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			
Series mode inputs SER_MD_2 SER_MD_1 SER_MD_0	INIT/RST	ON	n.a.	Pull Down ON	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			
External regulator status REGS	INIT/RST	ON	n.a.	n.a.	
	RST	ON			
	RUN	ON			
	SLEEP	ON			
	STOP	ON			
	STOP-HIZ	ON			

Pins / Ports	Device state	Pin states			Comments
		Input state	Output state	Pull-Up/Down	
External regulator control REGC_3 REGC_2 REGC_1 REGC_0	INIT/RST	n.a.	HiZ	n.a.	
	RST		enabled		Open Drain outputs
	RUN				
	SLEEP				
	STOP				
	STOP-HIZ				

MB91460 Series

Chapter 4 CPU Architecture

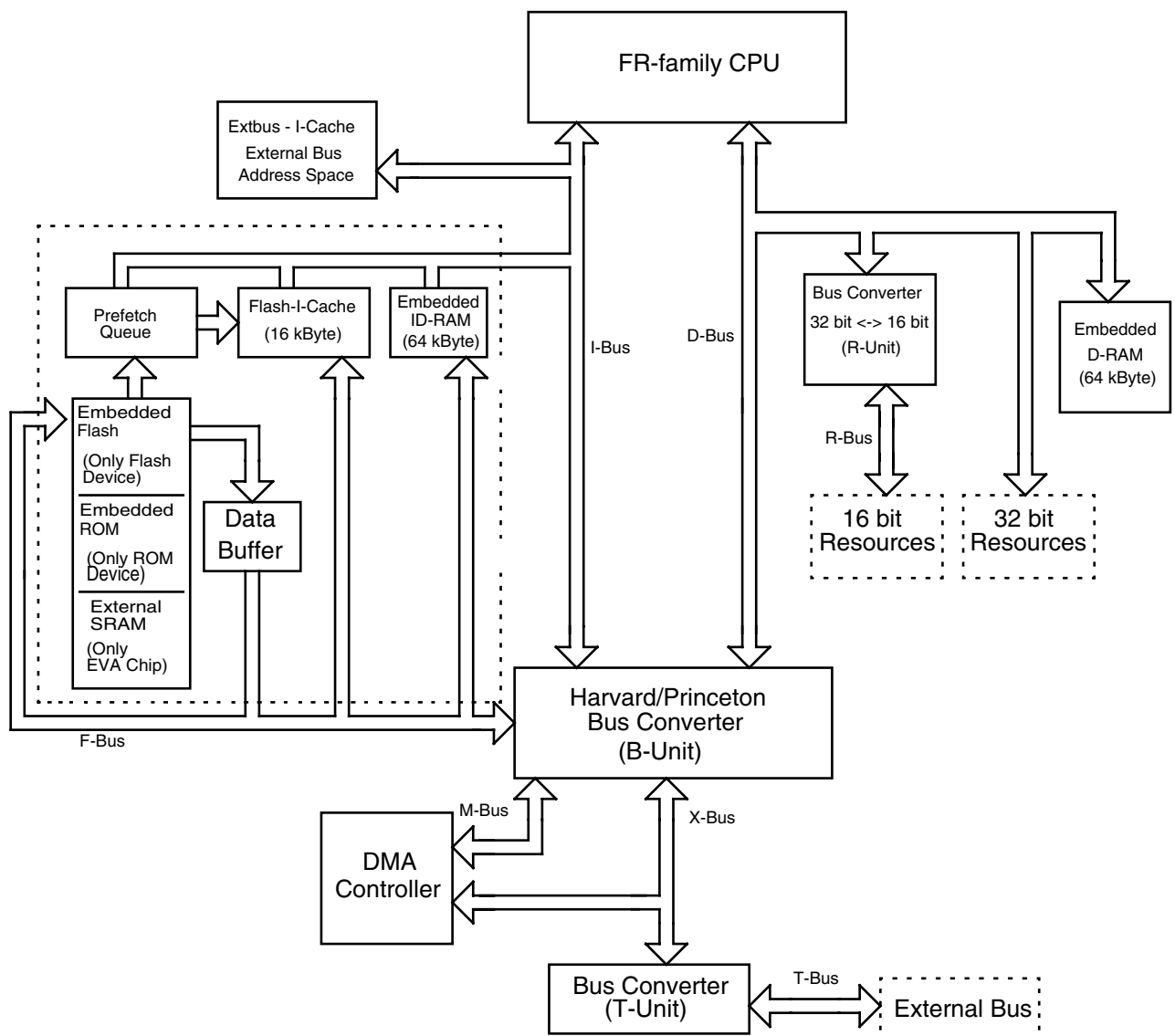
This chapter describes the architecture of FR-family CPU.

4.1. Overview

The CPUs of the FR-family series employ RISC architecture and advanced function instructions for embedded application.

CPU of FR-family employs Harvard architecture whose instruction bus and data bus are independent. "32-bit/16-bit bus converter" realizes the interface between CPU and peripheral functions. "Harvard/Princeton bus converter" connects both of I-bus and D-bus and realizes the interface between CPU and Princeton Bus.

Figure 4.1-1 Connection Diagram of Internal Architecture (Core Group)



4.2. Features

■ Features of internal architecture

- RISC architecture
- Base instruction: 1 instruction/1 cycle
- 32-bit architecture
- General-purpose register: 32-bit x 16
- 4GB of linear memory space
- Equipped with multiplier.
 - 32-bit x 32-bit multiplication: 5 cycles
 - 16-bit x 16-bit multiplication: 3 cycles
- Enhanced interrupt processing function
 - High-speed respond (6 cycles)
 - Support of multiple interrupts
 - Level mask function (16 levels)
- Enhanced instruction for I/O operation
 - Transfer instruction between memories
 - Bit-processing instruction
- Highly efficient code
- Length of base instruction words: 16 bits
- Standby states (Low power consumption states): SLEEP/STOP
- Setting function of clock division ratios

MB91460 Series

4.3. CPU

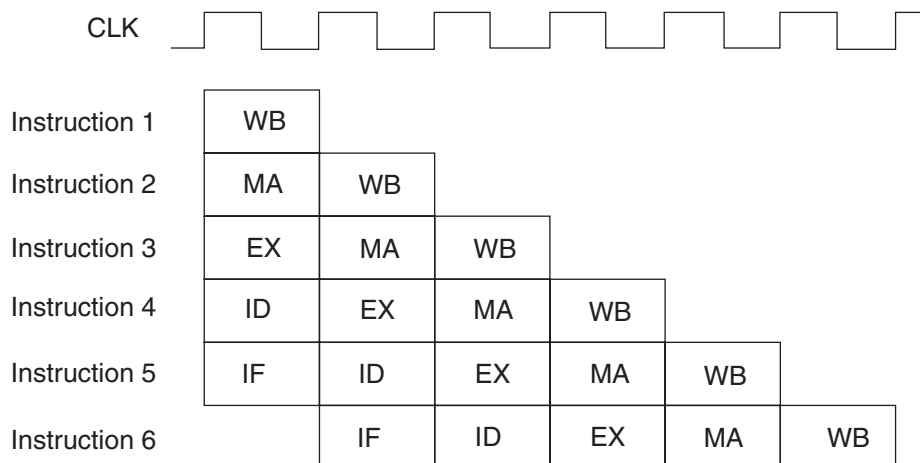
The CPU realizes the compact implementation of a 32-bit RISC FR architecture.

It employs a 5-stage instruction pipeline method to execute 1 instruction per 1 cycle.

This pipeline consists of the following stages.

- Instruction fetch (IF): outputs instruction address to fetch instruction.
- Instruction decode (ID): decodes fetched instruction and reads register.
- Execution (EX): executes operation.
- Memory access (MA): loads data for memory or accesses stored data.
- Write back to register (WB): writes back data to registers.

Figure 4.3-1 Instruction Pipeline



No instruction is executed in random order. If instruction A enters into pipeline before instruction B, instruction A always reaches to write-back stage before instruction B.

1 instruction is executed per 1 cycle.

However, to execute the instruction, multiple cycles are required for load/store instruction with memory wait, branch instruction without delay slot and multi-cycle instruction. In addition, a slow instruction degrades instruction execution speed.

4.4. 32-bit/16-bit Bus Converter

This converter generates the interface between D-bus which executes 32-bit high-speed access and R-bus which executes 16-bit access in order to realize data access from CPU to peripheral functions.

If 32-bit access comes from CPU, this converter converts the access into two 16-bit accesses to access to R bus. Some peripheral functions have restrictions of access width.

4.5. Harvard/Princeton Bus Converter

This converter realizes interface between instruction access and data access of CPU, to realize smooth interface with external bus.

The CPU employs the Harvard architecture whose instruction bus and data bus are independent while it employs single-bus Princeton architecture for bus controller to control external bus. This bus converter prioritizes instruction accesses and data accesses of CPU, and executes access control to bus controller. This always optimizes access sequence to external bus.

4.6. Instruction Overview

The FR-family supports logic operation, bit operation and direct addressing instruction optimized for embedded application as well as general RISC instruction system. Instruction-set list is shown in the appendix. Since each instruction is 16-bit length (some instruction is 32-bit or 48-bit length), it is possible to generate compact program code.

Instruction sets are grouped into the following a through f function groups.

4.6.1 Arithmetic Operation

This group consists of standard arithmetic operation instructions (addition, subtraction and comparison) and shift instruction (logic shift and arithmetic shift). For addition and subtraction, the operation with carry used for multiple word length operation and the operation useful for address calculation without changing flag value are allowed.

In addition, it includes 32-bit x 32-bit and 16-bit x 16-bit multiplication instruction as well as 32-bit/32-bit step division instruction.

It provides transfer instruction of immediate value which sets immediate value to register, and transfer instruction between registers.

All arithmetic instructions are operated using general-purpose register and multiply & divide register within CPU.

4.6.2 Load and Store

Load/store is the instruction to read and write to memory. This is also used for read and write to peripheral functions (I/O) within chip.

Load and store consist of 3 type access lengths including byte, half-word and word. In addition to general register-indirect memory addressing, some instructions allow register-indirect memory addressing with displacement or with register increment/decrement.

4.6.3 Branch

This is the instruction for branch, call, interrupt and return. Branch instruction consists of instructions with and without delay slot. For more information of branch instruction, see [“Chapter 7 Branch Instruction \(Page No.227\)”](#).

4.6.4 Logical Operation and Bit Operation

Logical operation instruction allows the logical operation of AND, OR and EOR between general-purpose registers or general-purpose register and memory (and I/O). Bit operation instruction allows the direct operation of data of memory (and I/O). Memory addressing is general register indirect.

4.6.5 Direct Addressing

Direct addressing instruction is the instruction to access between I/O and general-purpose register, or between I/O and memory. By directly instructing I/O address rather than register indirect, it enables high-speed and high-efficient access. Some instructions allow register-indirect memory addressing with register increment/decrement.

4.6.6 Others

This is the instruction which executes flag setting, stack operation, sign extension and zero extension within PS register. It provides the function entrance/exit which supports high-level language, and register multi-load/store instruction.

MB91460 Series

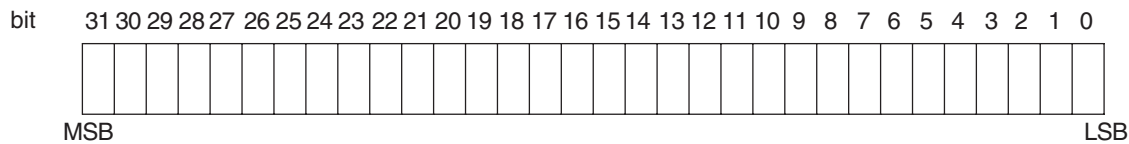
4.7. Data Structure

FR-family CPU has two data allocations as follows.

4.7.1 Bit Ordering

FR-family CPU uses little endian as bit ordering.

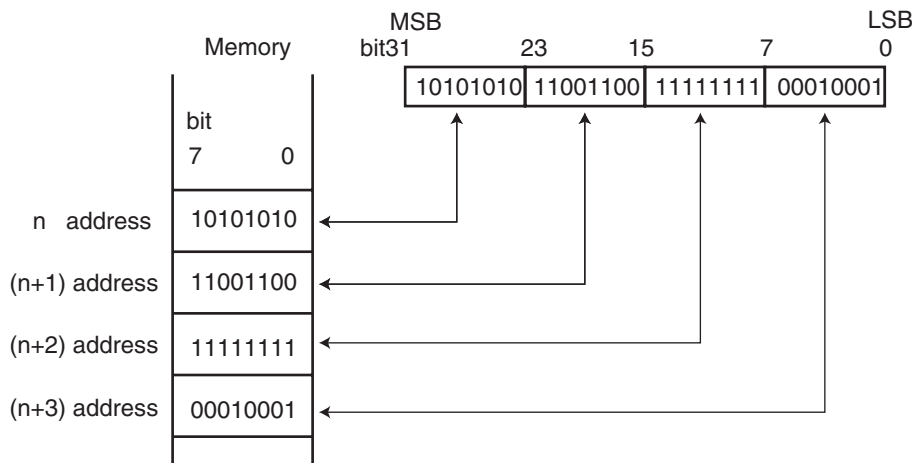
Figure 4.7-1 Bit Structure of Bit Ordering



4.7.2 Byte Ordering

FR-family CPU uses big endian as byte ordering.

Figure 4.7-2 Bit Structure of Byte Ordering



4.8. Word Alignment

Since instructions and data are accessed by byte, allocated addresses vary by instruction length or data width.

4.8.1 Program Access

FR-family CPU program is required to be allocated in addresses multiplied by 2.

PC's bit0 is cleared for instruction execution upon the PC update.

(PC bit 0 may be set when odd address is specified for branching address, however, it is invalid. Since the instruction is required to be allocated in addresses multiplied by 2, there is no odd address exception.)

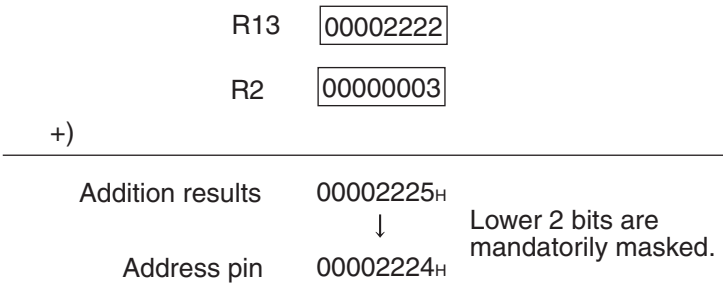
4.8.2 Data Access

FR-family CPU provides the following alignment for addresses depending on data width when executing data access.

- Word access : Address is multiplied by 4. (Lowest order 2 bits are mandatorily 00.)
- Half-word access : Address is multiplied by 2. (Lowest order bit is mandatorily 0.)
- Byte access: : Address is multiplied by 1.

Upon the word and half-word data accesses, some bits mandatorily become 0 for computing results of effective address. For example, in the case of addressing mode of @(R13, R_i), register value before addition is used as is (even if lowest order bit is 1), and lower bits of addition results are masked. Register values before computing are not masked.

Figure 4.8-1 [Example] LD @(R13, R2), R0



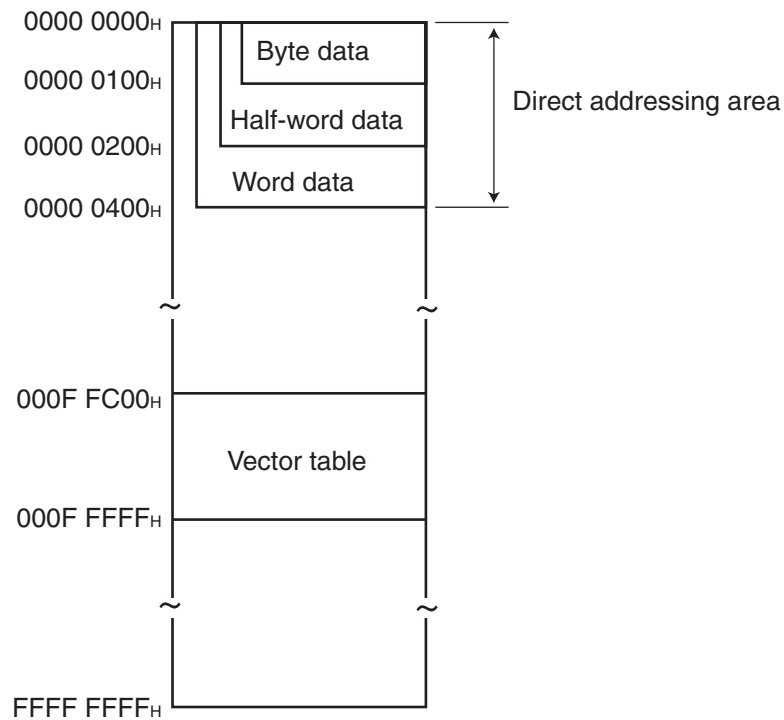
MB91460 Series

4.9. Addressing

Address space is 32-bit linear.

4.9.1 Map of address space

Figure 4.9-1 Address Map



FR-family CPU has logical address space of 4GB (2^{32} addresses), CPU accesses the data linearly.

4.9.2 Direct Addressing Area

The following areas are used for I/O.

These spaces are referred to as direct addressing area where direct operand address can be specified by the instruction.

These direct areas vary by data size to be accessed.

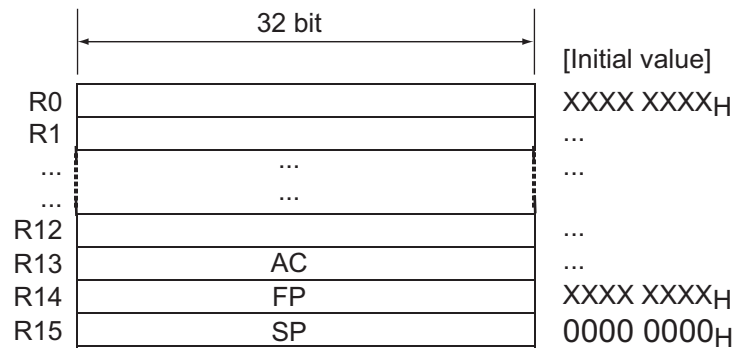
- Byte access : 0 - 0FF_H
- Half-word access : 0 - 1FF_H
- Word access : 0 - 3FF_H

Chapter 5 CPU Registers

5.1. General-purpose Registers

Registers R0 through R15 are general-purpose registers. These registers are used for accumulator and memory access pointers on various operations.

Figure 5.1-1 General-purpose Registers



Of 16 registers, the following registers are reserved for special application.(In addition to functioning as general-purpose registers)

- R13: Virtual accumulator
- R14: Frame pointer
- R15: Stack pointer

Initial values by reset are indeterminate for R0 through R14. Initial value by reset is 00000000_H (SSP value) for R15.

5.2. Dedicated Registers

Dedicated registers consist of program counter (PC), program status (PS), table-base register (TBR), return pointer (RP), system stack pointer (SSP), user stack pointer (USP) and multiply & divide register (MDH/MDL).

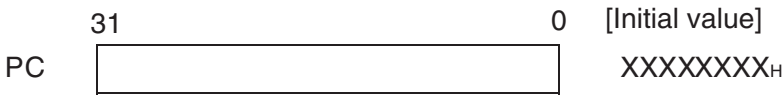
Figure 5.2-1 Dedicated Registers List

Program Counter	(PC)	<div></div>
Program status	(PS)	<div><div>-</div><div>ILM</div><div>-</div><div>SCR</div><div>CCR</div></div>
Table-base register	(TBR)	<div></div>
Return pointer	(RP)	<div></div>
System stack pointer	(SSP)	<div></div>
User stack pointer	(USP)	<div></div>
Multiply & divide registers	(MDH)	<div></div>
	(MDL)	<div></div>

5.2.1 PC: Program Counter

Program Counter (PC) consists of 32 bits.

Figure 5.2-2 Bit Structure of Program Counter (PC)



This register indicates the address of the instruction that is currently executing. After a reset, the initial value of the PC is the reset entry address contained in the vector table.

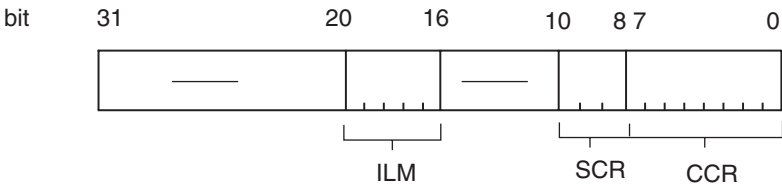
5.2.2 PS: Program Status Register

Program status register (PS) is the register to hold program status which consists of three parts including ILM, SCR and CCR.

All undefined bits are reserved bit. Upon the reading, “0” is always read. Writing is invalid.

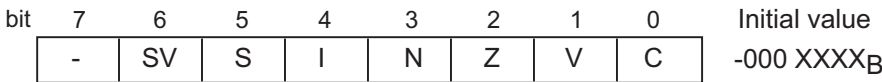
Program status register (PS) consists of condition code register (CCR), system condition code register (SCR) and interrupt level mask register (ILM).

Figure 5.2-3 Bit Structure of Program Status (PS)



■ CCR: Condition Code Register

Figure 5.2-4 Structure of Condition Code Register (CCR)



- [Bit 6] SV: SuperVisor flag

This bit indicates supervisor mode of the CPU/MPU

SV	Description
0	CPU/MPU is in User Mode
1	CPU/MPU is in Supervisor Mode

This bit becomes “0” by reset.

Signals SVMODE to the EDSU/MPU. Can be triggered by the INT #5 instruction (sets the SV flag and executes INT#5 ISR) or by using ORCCR to set, resp. ANDCCR to clear the SV-flag. Hardware behaviour by execution of INT #5 is same as executing other INT instructions except that the SV flag is set.

For the complete overview about the Memory Protection Unit (MPU) please refer to [Chapter 29 MPU / EDSU \(Page No.531\)](#).

MB91460 Series

- [Bit 5] S: Stack flag

This bit specifies stack pointer.

S	Description
0	Uses R15 as SSP. Upon generating EIT, this bit automatically becomes "0". (Note that the value saved in stack is the value before clear.)
1	Uses R15 as USP.

This bit becomes "0" by reset.

After using R15 as USP, write "0" before executing RETI instruction.

- [Bit 4] I: Interrupt enable bit

This bit enables and disables user interrupt request.

I	Description
0	Disables user interrupt.
	Upon executing INT instruction, this bit becomes "0".
	(Note that the value saved in stack is the value before clear.)
1	Enables user interrupt ¹ . Mask processing of user interrupt request is controlled by the value which is held in ILM.

1. For more information about user interrupt refer to "[Chapter 6 EIT: Exceptions, Interrupts and Traps \(Page No.217\)](#)"

This bit becomes "0" by reset.

- [Bit 3] N: Negative flag

This bit indicates the sign when operation results is deemed as integer represented by two's-complement numbers.

N	Description
0	It indicates that operation result is positive value.
1	It indicates that operation result is negative value.

- [Bit 2] Z: Zero flag

It indicates whether operation result is 0 or not.

Z	Description
0	It indicates that operation result is other than 0.
1	It indicates that operation result is 0.

- [Bit 1] V: Overflow flag

This bit deems that operand used for operation as integer represented by two's-complement numbers, and indicates whether overflow was generated or not as the result of operation results.

V	Description
0	It indicates that overflow was not generated as the result of operation.
1	It indicates that overflow was generated as the result of operation.

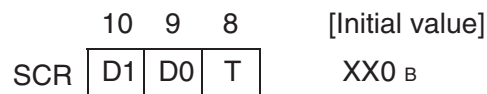
- [Bit 0] C: Carry flag

This bit indicates whether carry or borrow from highest-order bit was generated or not as the result of operation.

C	Description
0	It indicates that neither carry nor borrow is generated.
1	It indicates that either carry or borrow is generated.

■ SCR: System Condition Code Register

Figure 5.2-5 Structure of System Condition Code Register (SCR)



This section describes each bit structure of system condition code register (SCR).

- [Bit 10, 9] D1 and D0: Step division flag

D1 and D0 bits hold intermediate data during the execution of step division.

Do not modify data during the execution of division processing.

If other processes are executed during the execution of step division, step division is assured to be restarted by saving and returning PS register value.

Initial status by reset is indeterminate for D1 and D0 bits.

Upon executing DIV0S instruction, these bits are set by referring to dividend and divisor.

Upon executing DIV0U instruction, these bits mandatorily become "00".

- [Bit 8] T: Step trace trap flag

This bit is the flag to enable step trace trap or not.

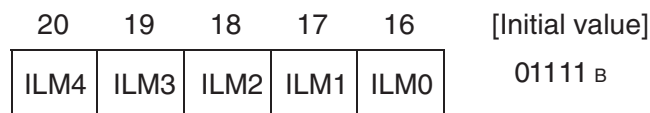
T	Description
0	Disables step trace trap.
1	Enables step trace trap. In this case, all user interrupts are disabled.

This bit is initialized to "0" by reset.

The function of step trace trap is used for emulator. During the use of emulator, this bit cannot be used for user program.

■ ILM: Interrupt Level Mask Register

Figure 5.2-6 Register Structure of Interrupt Level Mask Register (ILM)



- This is the register to hold interrupt level mask value. This bit uses the value held in ILM as level mask.

- ILM indicates corresponding interrupt level from interrupt requests entered in CPU.

- Interrupt requests are accepted only if it's priority is higher than the level.

- For level value, the highest priority is 0 (00000_B), and the lowest priority is 31 (11111_B).

- Program has some restrictions on configurable data.

- When original value is between 16 and 31,

Configurable new values are the value between 16 and 31.

If the value to be set is specified between 0 and 15, "specified value +16" is set.

- When original value is between 0 and 15,

Any value between 0 and 31 can be set.

These values are initialized to 15 (01111_B) by reset.

MB91460 Series

■ Caution: PS Register

Since some instructions have already processed PS register in advance, the following exception operations may break interrupt processing routine during the use of debugger, or update PS flag data.

In either cases, after returning from EIT, it is designed to execute the correct process so that operations before and after EIT will be processed in accordance with specification.

• **At instruction right before DIV0U/DIV0S instruction, the following 1. to 3. operation may be executed.**

- If user interrupt request is received,
- If step execution is executed,
- If data event or emulator menu is broken,

1.D0 or D1 flag is updated in first.

2.EIT processing routine (user interrupt or emulator) is executed.

3.After returning from EIT, it executes DIV0U/DIV0S instruction and updates D0/D1 flag to the same value as 1.

• **When user interrupt request is generated, if any instruction of ORCCR, STILM, MOV Ri or PS is executed to enable interrupt, the following operations are generated.**

1.Updates PS register in first.

2.Executes EIT processing routine (user interrupt).

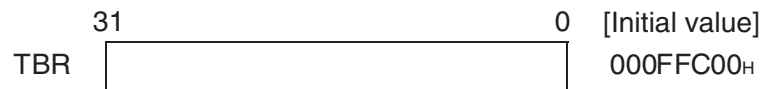
3.After returning from EIT, executes the instruction above and updates PS register to the same value as 1.

Note: For EIT, See “[Chapter 6 EIT: Exceptions, Interrupts and Traps \(Page No.217\)](#)”.

5.2.3 TBR: Table-base Register

Table-base register (TBR) consists of 32 bits.

Figure 5.2-7 Bit Structure of Table-base Register (TBR)

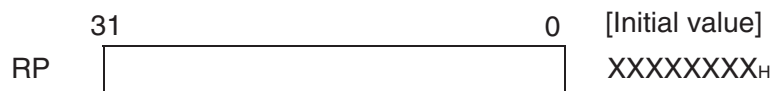


The Table-base register holds the first address of the vector table to be used during EIT processing.
The initial value upon reset is 000FFC00.

5.2.4 RP: Return Pointer

Return pointer (RP: Return Pointer) consists of 32 bits.

Figure 5.2-8 Bit Structure of Return Pointer (RP)



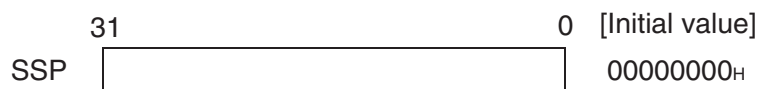
The return pointer holds the return address from a subroutine.
When the CALL instruction is executed, the value of the PC is transferred to the RP.
When the RET instruction is executed, the contents of the RP are transferred to the PC.
The initial value upon reset is undefined.

5.2.5 SSP: System Stack Pointer

System stack pointer (SSP) is used for the pointer which receives EIT and indicates stack to save/return data for return operation.

System stack pointer (SSP) consists of 32 bits.

Figure 5.2-9 Bit Structure of System Stack Pointer (SSP)



When S flag is "0", it works as R15. SSP can explicitly specified.

Upon generating EIT, it is used for the pointer which specifies the stack to save PS and PC.

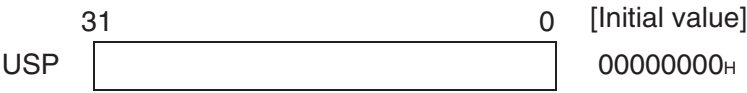
During the EIT process, this pointer reduces the value by 8, and adds 8 to the value during the return from EIT by executing RETI instruction.

System stack pointer (SSP) works as general-purpose register R15 when S flag within CCR is "0".

5.2.6 USP: User Stack Pointer

User Stack Pointer (USP) consists of 32 bits.

Figure 5.2-10 Bit Structure of User Stack Pointer (USP)



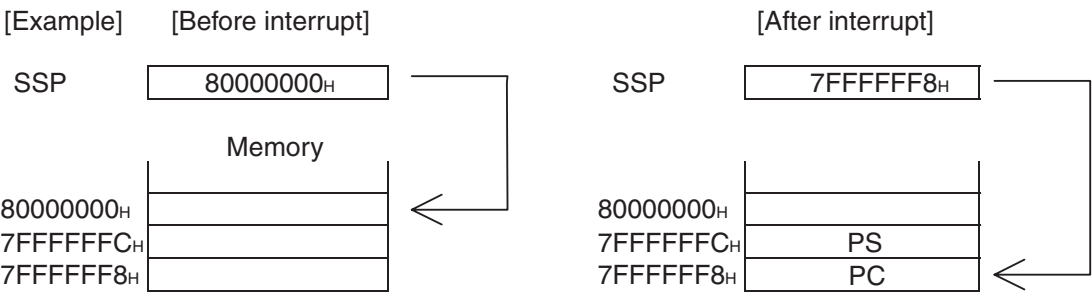
When S flag is “1”, this pointer works as R15.

USP can be explicitly specified.

RETI instruction cannot be used

This pointer saves and returns PC and PS values at the position where system stack pointer (SSP) indicates. After interrupt, it stores PC in address where SSP indicates, and PS in (SSP+4) address.

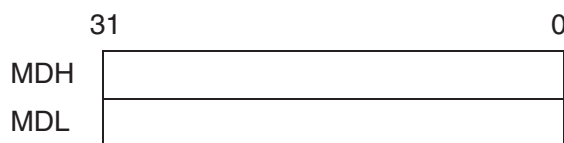
Figure 5.2-11 Interrupt Stack



5.2.7 MDH, MDL: Multiply & Divide Register

Multiply & Divide register (MDH/MDL) consists of 32 bits.

Figure 5.2-12 Bit Structure of Multiply & Divide Register (MDH/MDL)



This is the register for multiplication and division and consists of 32 bits.

Initial value by reset is indeterminate.

■ At the executing multiplication

When 32 bits x 32 bits multiplication, operation results of 64 bits are stored in multiplication/division store register as the following allocation.

- MDH: Upper 32 bits
- MDL: Lower 32 bits

When 16 bits x 16 bits multiplication, results are stored as follows.

- MDH: Indeterminate.
- MDL: Results of 32 bits

■ At the executing division

Upon starting operation, dividend is stored in MDL.

By computing division by executing DIV0S/DIV0U, DIV1, DIV2, DIV3, or DIV4S instruction, results are stored in MDL and MDH.

- MDH: Remainder
- MDL: Quotient

Chapter 6 EIT: Exceptions, Interrupts and Traps

6.1. Overview

EIT stands for Exception, Interrupt and Trap.

Interrupts, exceptions and traps are similar operations applied under partially differing conditions. Each "EIT" event involves terminating execution of instructions, saving information for restarting, and branching to a designated processing program. "EIT" processing programs can return to the prior program by use of the "RETI" instruction.

"EIT" processing operates in essentially the same manner for exceptions, interrupts and traps, with the following minor differences:

- **Interrupts**

originate independently of the instruction sequence. Processing is designed to resume from the instruction immediately following the acceptance of the interrupt.

- **Exceptions**

are related to the instruction sequence, and processing is designed to resume from the instruction in which the exception occurred.

- **Traps**

are also related to the instruction sequence, and processing is designed to resume from the instruction immediately following the instruction in which the trap occurred.

6.2. Features

- Support of nested EITs:
 - (multiple and simultaneous EITs)
- Both maskable and non-maskable interrupts
 - The priority of all maskable interrupts is lower than the non-maskable ones
 - Assignment of maskable interrupt priority levels by user
- Execution instructions
- Trap instructions
- Emulator operation support
- The priority of all EIT operations is lower than the Reset one

6.3. EIT Trigger

- Reset (highest priority of all)
- Interrupts
 - Maskable interrupts
 - Non-maskable interrupts
- Delayed interrupt
- Undefined-instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap
- Coprocessor absent trap (only on devices with mounted coprocessor)
- Coprocessor error trap (only on devices with mounted coprocessor)

6.4.Context saving

- CPU supervisor mode
- Memory protection violation

6.4. Context saving

In the case of EITs processing, the values of the "PS" and "PC" are saved to the stack as designated by the "SSP", regardless of the value of the "S" flag in the "CCR".

6.5. Recovery from EIT handler

RETI instruction is used for recovery from the EIT handler.

In order to guarantee the program execution results after recovery, it is required that all the contents of the CPU register are saved.

It is important to note that the PC and PS values in the stack are not overwritten unless needed because those values, saved in the stack at the occurrence of EIT, are recovered from the stack during the recovery sequence using the RETI instruction. Be sure to set the "S" flag to 0 when the RETI instruction is executed.

6.6. EIT Interrupt Level

When multiple "EIT" requests occur at the same time, priority levels are used to select one source and execute the corresponding "EIT" sequence. After the "EIT" sequence is completed, "EIT" request detection is applied again to enable processing of multiple "EIT" requests. Acceptance of certain types of "EIT" requests can mask other events. In such cases the priority applied by the "EIT" processing handler may not match the priority of the requests.

Table 6.6-1 Interrupt Level of EIT

Level		Description	Remarks
Binary	Decimal		
00000	0	(Reserved for system)	If original value of ILM is between 16 and 31, these values are not configurable to ILM by program.
...	
...	
00011	3	(Reserved for system)	
00100	4	INTE instruction Step trace trap	
00101	5	(Reserved for system)	
...	
...	
01110	14	(Reserved for system)	When ILM is set, user Interrupt is disabled.
01111	15	NMI (for users)	
10000	16	Interrupt	
10001	17	Interrupt	
...	
...	
11110	30	Interrupt	
11111	31	N/A	When ICR is set, Interrupt is disabled.

The non-maskable interrupt priority cannot be configured by user. This priority is 15.

The maskable interrupt priority can be set by user. The priority of each maskable interrupt can be set in the range from 16 to 31.

If the priority is equal to 31, then interrupt is disabled. This means that it will not be assigned.

Undefined-instruction exception, coprocessor absent trap, coprocessor error trap and INT instruction are not affected by interrupt level. Also, ILM is not changed by interrupt level.

6.7. EIT Vector Table

The vector table is located in the main memory, occupying an area of 1K bytes beginning with the address contained in the TBR. Vectors are 4 bytes in length. Therefore, 255 vectors are located in this area in the main memory.

These areas are intended for use as a table of entry addresses for "EIT" processing. However in applications where vector tables are not required, this area can be used as a normal instruction or data area.

The vector address is calculated as follows:

$$\text{Vector address} = [\text{TBR}] + \text{Offset value} = [\text{TBR}] + (03\text{FCH} - 4 \times \text{Vector number})$$

Lower two bits as the result of addition are always used for "00".

The initial value of the TBR register after a reset is equal to 000FFC00H. Therefore, the initial location of the vector table after a reset is 000FFC00H through 000FFFFFFH.

Both the mode and reset vector have a fixed addresses 000FFFF8H and 000FFFFCH respectively. These addresses remain fix, although the TBR register is rewritten.

For more information about the EIT vector table, refer to the chapter of “3.3. Interrupt Vector Table (Page No.113)”.

6.8. Multiple EIT Processing

If multiple EITs are generated at the same time, CPU repeats the operation which selects one of the EIT to accept, and then executes EIT sequence, and detects EIT again. If there is no EIT to accept upon detecting EIT, CPU executes instruction of the last accepted EIT handler. Therefore, if multiple EITs are generated at the same time, execution sequence of each EIT handler is determined by the following two parameters.

■ Basic Operations in "EIT" Processing

The FR family device processes "EIT" events as follows:

- (1) The vector table indicated by the table base register (TBR) and the number corresponding to the particular "EIT" event are used to determine the entry address for the processing program for the "EIT".
- (2) For restarting purposes, the contents of the old program counter (PC) and the old program status (PS) are saved to the stack area designated by the system stack pointer (SSP).
- (3) After the processing flow is completed, the presence of new "EIT" sources is determined.

■ Priority Level of EIT Triggers

Priority level of EIT triggers defines the sequence to select which EIT triggers to execute by saving PS and PC in order to update PC and masking other triggers where appropriate.

EIT does not always mean first-in first-out handler.

Table 6.8-1 Priority Level of Receipt of EIT Triggers and Mask for Other Triggers

Priority for accepting EITs	EIT	Masking of other EITs
1	Reset	All EITs are cleared
2	Instruction Break	Other EITs are canceled (ILM = 4)
3	INTE instruction	Other EITs are canceled (ILM = 4)
4	Undefined instruction exception	Other EITs are canceled (I-flag = 0)
5	INT instruction / Coprocessor exceptions	I-flag = 0
6	Memory protection violation	I-flag = 0
7	User interrupt	ILM = level of accepted INT
8	NMI (user)	ILM = 15
9	NMI (emulator)	Other EITs are canceled (ILM = 4)
10	Step Trace trap	Other EITs are canceled (ILM = 4)
11	Operand Break	Other EITs are canceled (ILM = 4)

■ How to Mask Other Triggers upon acceptance

Table below shows the execution sequence of each handler when multiple EIT requests are generated at the same time considering mask processes.

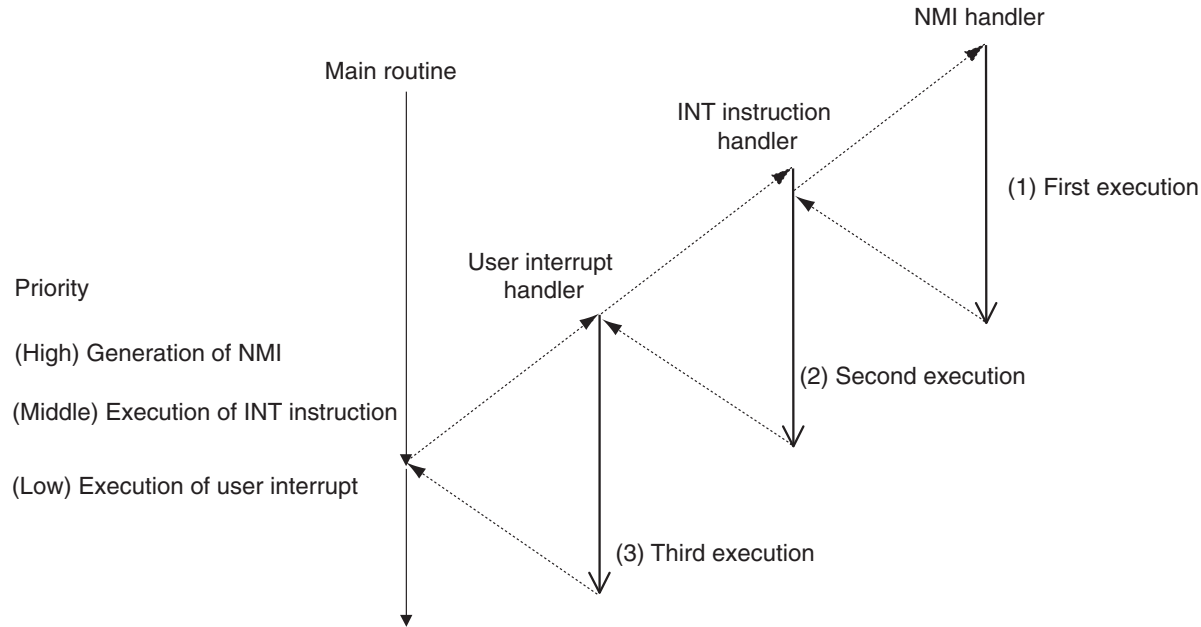
Table 6.8-2 Execution Sequence of EIT Handlers

Handler execution Priority	EIT	Masking of other EITs
1	Reset	All EITs are cleared
2	Undefined instruction exception	Other EITs are canceled (I-flag = 0)
3	Instruction Break	Other EITs are canceled (ILM = 4)
4	INTE instruction	Other EITs are canceled (ILM = 4)
5	NMI (emulator)	Other EITs are canceled (ILM = 4)
6	Step Trace trap	Other EITs are canceled (ILM = 4)
7	Operand Break	Other EITs are canceled (ILM = 4)
8	NMI (user)	ILM = 15

Table 6.8-2 Execution Sequence of EIT Handlers

9	Memory protection violation	I-flag = 0
10	INT instruction / Coprocessor exceptions	I-flag = 0
11	User interrupt	ILM = level of accepted INT

Figure 6.8-1 Multiple EITs Process



6.9. Operation

In this section, the following nomenclature is used:

Source "PC" is the address of the current instruction which is executing at the time when an EIT request takes place.

"Address of next instruction" is the next address from the current instruction which is executing at the time when an EIT request takes place.

- When LDI is 32: PC+6
- When LDI is 20, and it is COPOP, COPLD, COPST or COPSV: PC+4
- For other instructions: PC+2

6.9.1 User Interrupt operation

If user interrupt request occurs, it determines whether to accept its request or not in the following sequence.

■ How to determine whether to accept interrupt request or not

1. Selects the interrupt which holds the highest priority level (the smallest number) by comparing interrupt request levels generated at the same time.
For the level to be compared, it uses the value which ICR holds corresponding to maskable interrupt.
2. Selects the interrupt request which has the lowest interrupt number if multiple interrupt requests with the same priority level are generated.
3. Where "Interrupt level \geq Level mask value", the interrupt request is masked without acceptance. Where "Interrupt level $<$ Level mask value", it goes forward to Step 4.
4. When selected interrupt request is maskable interrupt, if I flag is 0, its Interrupt request is masked without acceptance and if I flag is 1, it goes forward to Step 5.
5. If the conditions above are satisfied, interrupt requests are accepted between instruction processes.
If user interrupt requests are accepted upon detecting EIT requests, CPU executes the following operations according to the Interrupt number for the interrupt request accepted.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the next instruction is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value (level) of the accepted interrupt is stored in the "ILM".
5. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
6. The vector address of the accepted interrupt is stored in the program counter (PC).

After the interrupt sequence, EIT is checked again before executing the main program handler's instruction. If any EIT is generated at this time, the CPU goes to EIT process sequence.

6.9.2 Operation of INT Instruction

INT #u8 instruction is operated as follows.

Branches to interrupt handler of vector specified in INT #u8.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the next instruction is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value "0" is written to the "I" flag in the condition code register (CCR) in the program status (PS).
5. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
6. The value 'TBR+3FC_H-4 x u8' is stored in the program counter (PC).

6.9.3 Operation of INTE Instruction

INTE instruction is operated as follows.

Branches to vector interrupt handler of vector #9.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the next instruction is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value (level) "4" is stored in the "ILM".
5. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
6. The value 'TBR+3D8_H' is stored in the program counter (PC).

During the execution of step, EIT is not generated by INTE.

Since INTE instruction is used for the debug support unit (DSU), do not use it.

6.9.4 Operation of Step Trace Trap

If T flag is set in SCR within PS and enable step trace trap function, step trace trap is generated with each following executing instruction.

■ Condition for detecting step trace trap

T flag = 1

Instructions are other than delayed branch command.

During the execution of instructions other than INTE instructions or step trace trap process routines.

If conditions above are satisfied, it is broken between instruction operations.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the next instruction is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value (level) "4" is stored in the "ILM".
5. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
6. The value 'TBR+3CC_H' is stored in the program counter (PC).

If T flag is set to enable step trace trap, user interrupt is disabled.

In addition, EIT will not be generated by INTE instruction.

FR-family CPU generates traps from next instruction to instruction which set T flag.

6.9.5 Operation of Undefined-instruction Exception

If any undefined instruction is detected upon decoding instruction, an undefined-instruction exception is generated.

■ Condition for detecting an undefined-instruction exception

- Upon the decoding instruction an undefined instruction is detected.
- It is out of delayed slot. (It is not the instruction which is right after delay branch instruction.)

If conditions above are satisfied, undefined-instruction exception will be generated.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the instruction that caused the undefined instruction exception is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
5. The value 'TBR+3C4_H' is stored in the program counter (PC).

The address of the instruction which has detected an undefined-instruction exception is saved as PC.

6.9.6 Coprocessor Absent Trap

If coprocessor instruction for unmounted coprocessor is executed, a coprocessor absent trap is generated.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the instruction following the one which caused the undefined instruction exception is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
5. The value 'TBR+3E0_H' is stored in the program counter (PC).

6.9.7 Coprocessor Error Trap

If an error occurs during the use of a coprocessor, a coprocessor error trap is generated when coprocessor instruction is executed in order to operate the coprocessor next time.

■ Operation

1. The contents of the program status (PS) are saved to the system stack.
2. The address of the instruction following the one which caused the undefined instruction exception is saved to the system stack.
3. The value of the system stack pointer (SSP) is reduced by 8.
4. The value "0" is written to the "S" flag in the condition code register (CCR) in the program status (PS).
5. The value 'TBR+3DC_H' is stored in the program counter (PC).

6.9.8 Operation of RETI Instruction

The RETI instruction is the instruction which returns from EIT process routine.

■ Operation

1. Load data from stack indicated by (R15)* to the program counter (PC).
2. Increment R15+4 and store to R15.
3. Load data from stack indicated by (R15)* to the program status (PS).
4. Increment R15+4 and store to R15.

The RETI instruction should be executed with S flag "0".

6.10. Caution

- Since the INTE instruction is used for the Debug Support Unit (DSU), do not use it in any application.
- The delay slot for branch instruction has restrictions on EIT.

See "[Chapter 7 Branch Instruction \(Page No.227\)](#)".

MB91460 Series

Chapter 7 Branch Instruction

7.1. Overview

The FR-family CPU can execute both delayed and non-delayed branching instructions.

The position of an instruction immediately following a branching instruction which is already loaded by the pipeline operation is called the delay slot.

In pipeline operation, by the time the CPU recognizes an instruction as a branching instruction the next instruction has already been loaded. To process the program as written, the instruction following the branching instruction must be canceled in the middle of execution. Branching instructions that are handled in this manner are non-delayed branching instructions. As a result, the program is processed in the order in which it is written, and the branching instruction requires an apparent processing time of two cycles.

A delayed branching instruction is a branching instruction that executes the instruction in the delay slot regardless of whether the branching conditions are satisfied or not satisfied. This means that the apparent order of instruction processing is changed.

The FR family CPU processes delayed branching instructions with an apparent execution speed of 1 cycle, regardless of whether branching conditions are satisfied or not. When branching occurs, this is one cycle faster than using non-delayed branching instructions.

7.2. List of Delayed Branching instructions

- The following branching instructions are delayed

JMP:D @Ri	CALL:D label12	CALL:D @Ri	RET:D
BRA:D label9	BNO:D label9	BEQ:D label9	BNE:D label9
BC:D label9	BNC:D label9	BN:D label9	BP:D label9
BV:D label9	BNV:D label9	BLT:D label9	BGE:D label9
BLE:D label9	BGT:D label9	BLS:D label9	BHI:D label9

7.3. Operation of Delayed Branching Instructions

In pipeline operation, the instruction located in the next position of a branching instruction is executed before. This means that just after the instruction located in the delay slot is executed, the branching instruction is processed. So that the apparent order of instruction processing is changed in cases where delayed branching occurs.

If a delay slot contains a not valid instruction, then a NOP instruction is executed.

Below the example shows how a delayed branching instruction is executed:

Figure 7.3-1 Example (Branch instruction with delay slot)

```

;      Sequence of instruction
ADD   R1,   R2   ;
BRA:D LABEL     ; Branch instruction
MOV   R2,   R3   ; Delay slot ..... To be executed before the branch.
...
LABEL: ST      R3,   @R4 ; Branched instruction

```

In case of conditional delayed branching instructions, the instruction located in the delay slot is executed regardless of whether the branching conditions are satisfied or not.

In delay branch instruction, execution sequence of some instructions seems opposite, however, it only applies to updating process on the PC. Any other operation (register update/look-up) is executed in the order of description.

7.4. Some Examples of Delayed Branching Instructions

7.4.1 JMP:D @Ri / CALL:D @Ri

Ri referred in JMP:D @Ri / CALL:D @Ri instruction remains intact even if instructions within delay slot update Ri.

Figure 7.4-1 Example

```
LDI:32  #Label,   R0
JMP:D   @R0       ; Branches to Label.
LDI:8    #0,      R0 ; Not affect any branched address.
...
```

7.4.2 RET:D Instruction

RP referred in RET:D instruction remains intact even if instructions within delay slot update RP.

Figure 7.4-2 Example

```
RET:D           ; Branches to the address previously specified in RP.
MOV    R8,      RP ; Not affect any return operation.
...
```

7.4.3 Bcc:D rel Instruction

Flag referred in Bcc:D rel instruction also remains unaffected by instructions within delay slot.

Figure 7.4-3 Example

```
ADD    #1,      R0 ; Change of flag
BC:D   Overflow   ; Branches in accordance with the execution result of instructions above.
ANDCCR #0        ; This flag update is not referred in branch instruction above.
...
```

7.4.4 CALL:D Instruction

When RP is referred using the instruction within delay slot of CALL:D instruction, the data updated by CALL:D instruction is read out.

Figure 7.4-4 Example

```
CALL:D Label    ; Branches by updating RP.
MOV    RP,      R0 ; Transfers RP based on the execution results of CALL: D above.
...
```

7.5. Restrictions on Branch Instruction with Delay Slot

7.5.1 Available Instructions for Delay Slot

Only the instructions located at the delay slot which meet the following requirements can be executed:

- **1-cycle instruction;**
- **Non-branch instruction; and**
- **Instruction which does not affect any operation even if its sequence is changed.**

"1-cycle instruction" indicates instructions whose number of cycles column in the instruction list table is described with "1", "a", "b", "c" or "d".

7.5.2 Step Trace Trap

The step trace trap is not generated at the time between the execution of a delayed branching instruction and the execution of the instruction located at the delay slot.

7.5.3 Interrupt

Interrupts are not accepted at the time between the execution of delayed branching instruction and the instruction located at the delay slot.

7.5.4 Undefined-instruction Exception

If an undefined instruction exists in delay slot, then the undefined instruction-exception is not generated. In this case, undefined instruction works as NOP instruction.

7.6. Branch Instruction without Delay Slot

- The following branching instructions are non-delayed:

JMP @Ri	CALL label12	CALL @Ri	RET
BRA label9	BNO label9	BEQ label9	BNE label9
BC label9	BNC label9	BN label9	BP label9
BV label9	BNV label9	BLT label9	BGE label9
BLE label9	BGT label9	BLS label9	BHI label9

7.7. List of Non-Delayed Branch Instructions

In pipeline operation, the instruction located in the next position of a non-delayed branching instruction is never executed.

The example below shows how a non-delayed branching instruction is executed:

Figure 7.7-1 Example (Branch Instruction without Delay Slot)

```

;      Sequence of instruction
ADD    R1,    R2    ;
BRA:D   LABEL      ; Branch instruction (without delay slot)
MOV     R2,    R3    ; Not to be executed.
...
LABEL   ST      R3,    @R4 ; Branched instruction

```

The number of execution cycles of non-delayed conditional branching instruction is equal to 2 cycles if the branching conditions are satisfied and only equal to 1 cycle if the branching conditions are not satisfied.

Therefore, select the delayed branching instruction when a valid instruction can be set at the delay slot. Otherwise select a non-delayed one in order to avoid an additional processing cycle regarding the execution of a NOP instruction instead of a non-valid instruction. This selection enables FR-family CPU to satisfy both of execution rate and code efficiency.

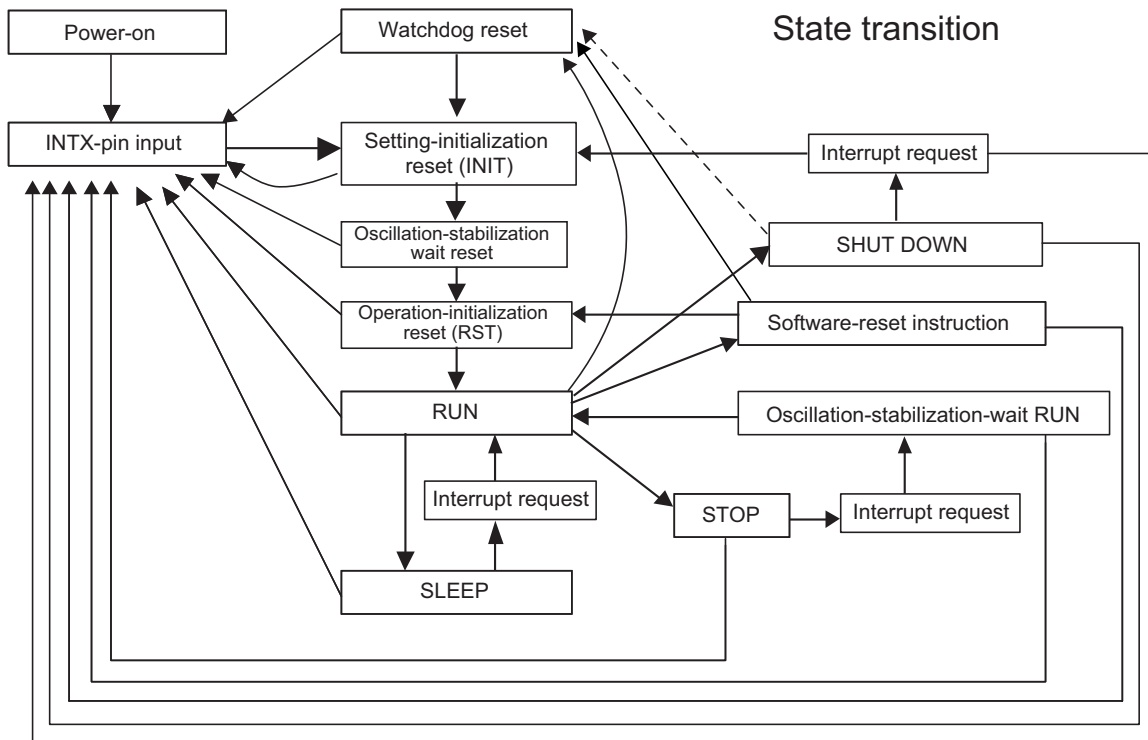
Chapter 8 Device State Transition

8.1. Overview

MB91460 basically has devices state and flow as shown below.

For more information, see “8.3. State Transition Diagram (Page No.232)”.

Figure 8.1-1 Overview Diagram of Device State Transition



8.2. Features

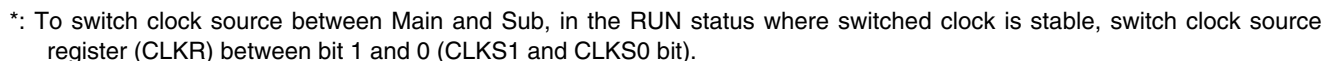
■ Device state

- RUN (Normal operation): State where the program is executed.
- SLEEP: State where the program is stopped (peripheral circuits are operating).
- STOP: State where the device is stopped.
- SHUT DOWN: State where the big part of the device and the main memories are switched off.
- Oscillation-stabilization-wait RUN: State to return from the STOP to the RUN state (waiting until clock oscillation is stabilized).
- Oscillation-stabilization-wait reset: State for waiting until the clock oscillation is stabilized after INIT.
- Operation-initialization reset (RST): State where the program is initialized.
- Setting-initialization reset (INIT): State where all settings are initialized.

■ Standby mode (Low-power-consumption mode)

SHUT DOWN, SLEEP and STOP above are standby modes.

Figure 8.3-1 State Transition of MB91460 Series



8.3.1 RUN (Normal Operation)

This is the state where program is executed with all clocks and all circuits are enabled.

This state has various paths for a state transition. However, if the synchronous reset mode is selected the state transition operations for some requests are different from normal reset mode. For more information, see the chapter of “[Chapter 9 Reset \(Page No.227\)](#)”.

8.3.2 SLEEP

This is the state where only the CPU's program execution is stopped and the peripheral circuits are operating. Embedded memories and internal/external buses are stopped unless the DMA controller requests them. This state is invoked by the program.

- Upon generation of valid interrupt requests the SLEEP state is cancelled and the RUN state (Normal operation) is entered.
- Upon generation of the external reset request by INITX pin the setting-initialization reset state (INIT) is entered.
- Upon generation of the operation-initialization reset request (by the external RSTX pin or software reset) the operation-initialization reset state (RST) is entered.

8.3.3 STOP

This is the state where all internal circuits, all internal clocks and the PLL are stopped. Main/Sub oscillation and RC oscillation (which can be connected to the Real Time Clock (RTC) can be stopped by setting the related registers). This state is invoked by the program.

Additionally high impedance for external pins can be enabled by setting the related register.

- Upon generation of specific valid interrupt requests (requiring no clock), active oscillation timer interrupt or Main clock oscillation stabilization timer interrupt request the oscillation stabilization wait RUN state is entered.
- Upon generation of the setting-initialization reset request by the external INITX pin the setting-initialization reset state (INIT) is entered.
- Upon generation of the operation-initialization reset request (by the external RSTX pin or software reset) the operation-initialization reset state (RST) is entered.

The Real Time Clock (RTC) can be supplied in the STOP state with either main or sub oscillation clock, if the control bits for oscillation disable (OSCDx of the STCR register) are not set to disable.

The Real Time Clock (RTC) can be supplied in the STOP state with CLKRC, if the control bit for oscillation enable (RCE of the CSVCR register) is not set to disable.

8.3.4 SHUT DOWN

This state is a special kind of STOP state for higher power saving. In this state the power supply of the big part of internal circuits and the main memories is switched off to minimize leakage. Main/Sub oscillation and RC oscillation (which can be connected to the Real Time Clock (RTC) can be stopped by setting the related registers). This state is invoked by the program (see [Chapter 64 ShutDown Control \(Page No.1631\)](#)).

High impedance for external pins can be enabled by setting the related register.

- Upon generation of specific valid external interrupt requests (requiring no clock), hardware watchdog timer interrupt or Real Time Clock (RTC) interrupt request the setting-initialization reset state (INIT) is entered.
- Upon generation of the setting-initialization reset request by the external INITX pin the setting-initialization reset state (INIT) is entered.

The Real Time Clock (RTC) can be supplied in the SHUT DOWN state with either main or sub oscillation clock, if the control bits for oscillation disable (OSCDx of the STCR register) are not set to disable.

The Real Time Clock (RTC) can be supplied in the SHUT DOWN state with CLKRC, if the control bit for oscillation enable (RCE of the CSVCR register) is not set to disable.

8.3.5 Oscillation-stabilization-wait RUN

All internal circuits are stopped except for clock generation control parts (timebase counter and device state control parts). All internal clocks are stopped while oscillation circuits and enabled Main PLL is operated. This state is entered automatically after the return from STOP.

- High-impedance control of external pins by STOP is cancelled.
- After the configured oscillation-stabilization-wait time has passed the RUN (Normal operation) state is entered.
- Upon generation of the setting-initialization reset request by the external INITX pin the setting-initialization reset state (INIT) is entered.
- Upon generation of the operation-initialization reset request (by the external RSTX pin or software reset) the operation-initialization reset state (RST) is entered.

8.3.6 Oscillation-stabilization-wait Reset

This is the state where the device is stopped. This state is entered upon a setting-initialization reset (INIT).

All internal circuits are stopped except for clock generation control parts (timebase counter and device state control parts). All internal clocks are stopped while oscillation circuits and Main PLL (if enabled) are operating.

- High-impedance control of external pins by STOP is cancelled.
- For internal circuits this state outputs operation-initialization reset (RST).
- After configured oscillation-stabilization-wait time has passed the oscillation-stabilization-wait reset state is entered.
- Upon generation of the setting-initialization reset request by the external INITX pin the setting-initialization reset state (INIT) is entered.

8.3.7 Operation-initialization Reset (RST)

This is the state where the program execution is initialized. Upon receipt of the operation-initialization reset (external RSTX pin or software reset) request or the termination of the oscillation-stabilization-wait reset (RST) this state is active.

CPU's program is stopped and program counter is initialized. All peripheral circuits are initialized except for some peripheral circuits. All internal clocks, oscillation circuits and enabled Main PLL are operating.

- For internal circuits this state asserts operation-initialization reset (RST).
- Upon clear of request of operation-initialization reset (RST) this state transits to the RUN (normal operation) state and executes the operation-initialization reset sequence. Upon returning from setting-initialization reset (INIT) this state executes the setting-initialization reset sequence.
- Upon generation of the setting-initialization reset request by the external INITX pin the setting-initialization reset state (INIT) is entered.

8.3.8 Setting-initialization Reset (INIT)

This is the state where all settings are initialized. Upon receipt of request of setting-initialization reset (INIT) this state is active.

CPU's program is stopped and program counter is initialized. All peripheral circuits are initialized. Oscillation circuits are operating while the Main PLL is stopped. All internal clocks are operating except while "L" level is input to the external INITX pin.

- For internal circuits this state asserts the setting-initialization reset (INIT) and the operation-initialization reset (RST).
- Upon clear of the setting-initialization reset (INIT) request this state cancels the setting-initialization reset state and then enters to the oscillation-stabilization-wait reset. After that it executes the operation-initialization reset sequence.

MB91460 Series**8.3.9 Priority of Each Request of State Transition**

In any state, each request of the state transition is subject to the following priority:

Highest priority	<ul style="list-style-type: none"> • Request of setting-initialization reset (INIT) • Termination of oscillation-stabilization-wait time (This is generated only in state of oscillation-stabilization-wait reset and oscillation-stabilization-wait RUN.)
↓	
↓	<ul style="list-style-type: none"> • Request of operation-initialization reset (RST)
↓	<ul style="list-style-type: none"> • Request of valid interrupt (This is generated only in RUN, SLEEP, STOP and SHUT DOWN state.)
↓	<ul style="list-style-type: none"> • Request of SHUT DOWN mode (Writing in register) (This is generated only in RUN state.)
Lowest priority	<ul style="list-style-type: none"> • Request of STOP mode (Writing in register) (This is generated only in RUN state.) • Request of SLEEP mode (Writing in register) (This is generated only in RUN state.)

MB91460 Series

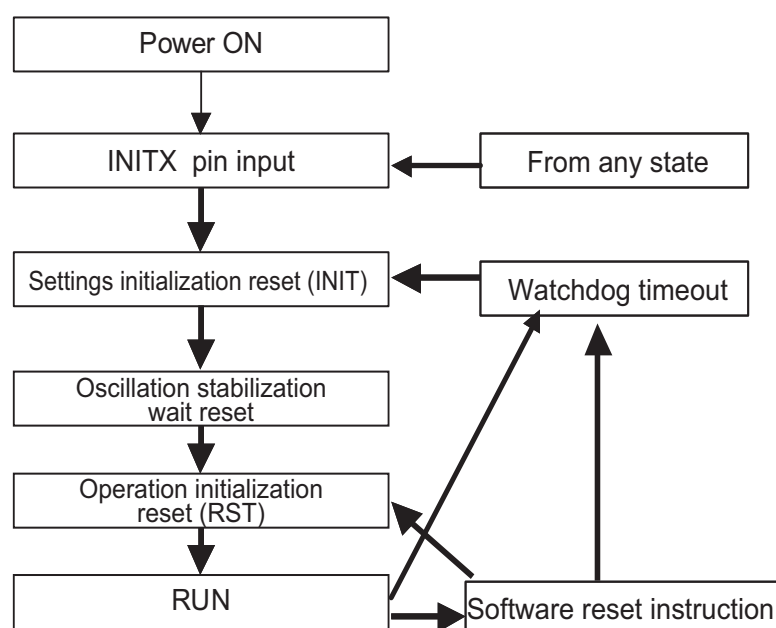
Chapter 9 Reset

9.1. Overview

When a reset is triggered, the device halts the program and all hardware operation, and then initializes all states. This state is called a reset.

When the reset trigger condition is removed, the device changes from this initialized state to restart the program and hardware operation. The series of steps from removal of the reset condition until operation starts is called the reset cancellation sequence.

Figure 9.1-1 Flow of Reset Operation



9.2. Features

9.2.1 Types of reset

- INITX pin input : Settings initialization reset (INIT)
- RSTX pin input : Operation initialization reset (RST)
- HSTX pin input : Settings initialization reset (INIT)
- Watchdog reset : Settings initialization reset (INIT)¹
- Software reset : Operation initialization reset (RST)
- Low Voltage reset : Settings initialization reset (INIT)
- Wake up from ShutDown : Settings initialization reset (INIT)²

In case of a clock supervisor reset in older devices (the device is running on CLKRC then), it is necessary to

1. Although a watchdog reset triggers the same settings initialization reset (INIT) as the INITX pin input, it does not initialize the oscillation stabilization time selection bits (OS[1:0]) and reset cause flags (INIT, HSTB, WDOG, ERST, SRST and LINIT).
2. In case of recovery from SHUT DOWN the special kind of setting initialization will be performed. See [64.4.4 Registers which are not initialized by Shutdwon Recovery \(Page No.1644\)\)](#)

have an external reset to start the device on CLKMAIN again. (not all INIT resets leads to this result).
See [Chapter 16 Clock Supervisor \(Page No.333\)](#).

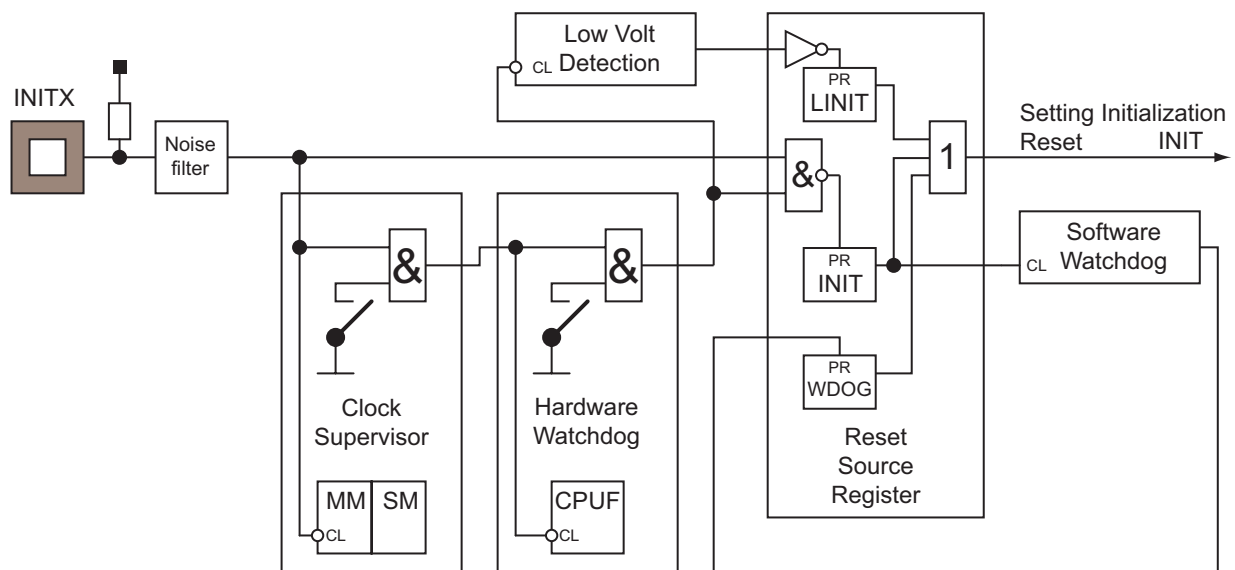
- **Cause of reset can be determined**

- The cause of the previous reset is stored in a series of flags (INIT, HSTB, WDOG, ERST, SRST and LINIT) in the RSRR register. See ["RSRR: Reset Cause Register" on P. 240](#) For the devices with SHUT DOWN mode see also [64.4.3 Determining the Reset Source after Shutdown \(Page No.1642\)](#).

9.2.2 External Reset, Watchdog Reset and Clock Supervisor Reset

The external INITX pin, the clock supervisor and the hardware watchdog built the external reset chain, which generates the Setting Initialization reset (INIT). Each module in the chain transfers the incoming reset signal to its reset output. External reset pin will clear all the modules in the chain, but the Hardware Watchdog reset will not clear the Clock Supervisor.

Figure 9.2-1 External reset / initialization chain



9.3. Configuration

Figure 9.3-1 Configuration Diagram of Reset operation

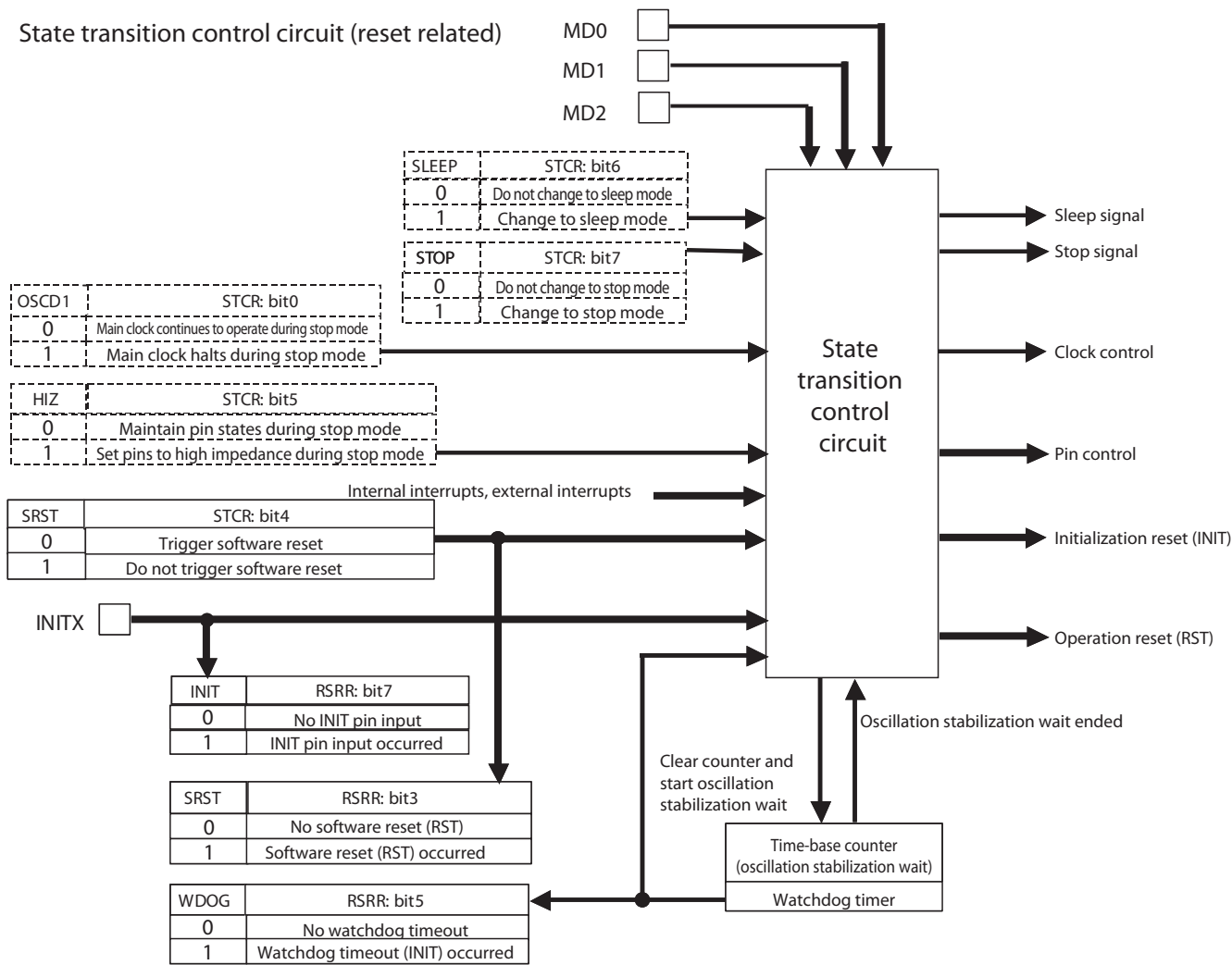


Figure 9.3-2 Register List

RSRR/STCR	
Address	
00480H	Bit 7
	6
00480H	5
	4
00480H	3
	2
00480H	1
	0
RSRR (Reset cause)	
00481H	
00481H	Bit 7
	6
00481H	5
	4
00481H	3
	2
00481H	1
	0
STCR (Standby control)	

9.4. Registers

9.4.1 RSRR: Reset Cause Register

Stores the cause of the previous reset, and sets the watchdog interval time for the software watchdog timer.

- **RSRR: Address 0480h (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
INIT	HSTB	WDOG	ERST	SRST	LINIT	WT1	WT0	
1	0	0	0	0	0	0	0	Initial value (INITX pin input)
-	-	-	X	X	-	0	0	Initial value (Watchdog reset)
X	X	X	-	-	X	0	0	Initial value (Software reset)
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	Attribute

Note: See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.

Reading the reset request cause returns the reset cause flags and then clears the flag values to “0”.

If multiple resets occur prior to reading the register, the resulting flag values contain the bitwise OR of the flags for each reset. That is, more than one flag may be set to “1”.

- Bit7: Initialization reset occurred flag

Indicates whether a reset (INIT) was triggered by INITX pin input.

INIT	Meaning
0	No INIT has been triggered by the INITX pin input.
1	INIT has been triggered by the INITX pin input or by the hardware watchdog.

The initialization reset occurred flag (INIT) is cleared to “0” after reading.

See “[Chapter 21 Hardware Watchdog Timer \(Page No.399\)](#)” for details.

Attention: The Clock Supervisor can set the INIT bit. See “[Check if reset was asserted by the Clock Supervisor](#)” on P. 354

- Bit6: Hardware Standby reset occurred flag

Indicates whether a reset (INIT) was triggered by HSTX pin input.

HSTB	Meaning
0	No INIT has been triggered by the HSTX pin input.
1	INIT has been triggered by the HSTX pin input.

The hardware standby reset occurred flag (HSTB) is cleared to “0” after reading.

- Bit5: Watchdog reset occurred flag

Indicates whether a reset (INIT) was triggered by the watchdog timer (SWWD).

WDOG	Meaning
0	No INIT has been triggered by the software watchdog timer.
1	INIT has been triggered by the software watchdog timer.

The watchdog reset occurred flag (WDOG) is cleared to “0” after reading.

- Bit4: External reset occurred flag

Indicates whether a reset (RST) was triggered by the RSTX pin input.

ERST	Meaning
0	No RST has been triggered by the RSTX pin input.
1	RST has been triggered by the RSTX pin input.

The external reset occurred flag (ERST) is cleared to “0” after reading.

MB91460 Series

- Bit3: Software reset occurred flag

Indicates whether a software reset has been triggered by writing to the software reset bit (STCR.SRST).

SRST	Meaning
0	No RST has been triggered by a software reset.
1	RST has been triggered by a software reset.

The software reset occurred flag (SRST) is cleared to “0” after reading.

- Bit2: Low voltage reset occurred flag

Indicates whether a reset (INIT) was triggered by the low voltage detection.

LINIT	Meaning
0	No INIT has been triggered by the low voltage detection.
1	INIT has been triggered by the low voltage detection.

The low voltage reset occurred flag (LINIT) is cleared to “0” after reading.

- Bit1-0: Watchdog interval time selection

The watchdog period selection bits (WT[1:0]) can set the period of the watchdog timer to the following:

($\phi \times 2^{20}$ to 2^{21} , $\phi \times 2^{22}$ to 2^{23} , $\phi \times 2^{24}$ to 2^{25} , $\phi \times 2^{26}$ to 2^{27})

See “[Chapter 20 Software Watchdog Timer \(Page No.389\)](#)” for details.

9.4.2 STCR: Standby Control Register

This register is used for software reset control (changing to standby mode, pin control in STOP state, and clock oscillation halted in STOP state), and specifies the oscillation stabilization wait time.

Note: See also "Chapter 10 Standby (Page No.241)".

• **STCR: Address 0481h (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	
0	0	1	1	0	0	1	1	Initial value (INITX pin input)
0	0	1	1	X	X	1	1	Initial value (Watchdog reset)
0	0	X	1	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R1,W	R/W	R/W	RX/WX	R/W	Attribute

Note: See "Meaning of Bit Attribute Symbols (Page No.15)" for details of the attributes.

• **Bit7: STOP state**

Writing "1" to the STOP state bit (STOP) changes to STOP state.

See "Chapter 10 Standby (Page No.241)" for details.

• **Bit6: SLEEP state**

Writing "1" to the SLEEP state bit (SLEEP) changes to SLEEP state.

See "Chapter 10 Standby (Page No.241)" for details.

• **Bit5: High impedance mode**

Writing "1" to the high impedance mode bit (HIZ) sets pin to high impedance (Hi-z) during STOP state.

See "Chapter 10 Standby (Page No.241)" for details.

• **Bit4: Software reset**

Writing "0" to the software reset bit triggers a software reset.

SRST	Operation
0	A software reset is issued.
1	No software reset is issued.

• Note that negative logic is used.

• The read value is always "1".

• **Bit3-2: Oscillation stabilization time selection**

The oscillation stabilization time selection bits (OS[1:0]) set the oscillation stabilization time as follows:

($\phi \times 2^1$, $\phi \times 2^{11}$, $\phi \times 2^{16}$, $\phi \times 2^{22}$)

The count is supplied by the timebase counter.

Initialized to "00" ($\phi \times 2^1$, Main clock) by a reset triggered by INITX pin input.

See "Chapter 19 Timebase Timer (Page No.379)" for details.

• **Bit1: Halt Sub clock oscillation**

Writing "1" to the halt Sub clock oscillation bit (OSCD2) halts the oscillation of the Sub clock during STOP state.

• **Bit0: Halt Main clock oscillation**

Writing "1" to the halt Main clock oscillation bit (OSCD1) halts the oscillation of the Main clock during STOP state.

MB91460 Series

9.4.3 MD: Mode Pins

These pins specify the location of the mode vector and reset vector that are read after the MCU is reset. The MD pins are validated by INITX pin = 0, other MD pins are not involved in the mode vector fetch.

Mode pins			Mode name	Reset vector Access area	Remarks
MD2	MD1	MD0			
0	0	0	Internal ROM mode vector	Internal	
0	0	1	External ROM mode vector	External	
1	0	1	debug mode		
1	1	1	Flash test / parallel programmer mode		
others			Reserved		

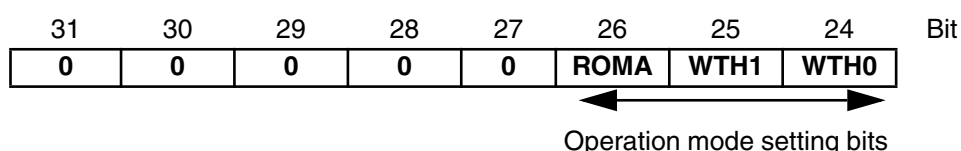
9.4.4 Mode Vector

The data written to the mode register (MODR) by the mode vector fetch operation is called the mode data. The mode register is an internal register and cannot be written to or read from directly. However, on MB91V460A and MB91FV460B, MODR can be read and written if the device is in emulation state.

After the mode register is set, the MCU operates in accordance with the modes (bus mode and access mode) set in this register.

The mode data is set by all types of reset. Setting the mode data from the user program is not possible.

- **Mode Vector: Address 000FFF8h (Access: Byte, Half-word, Word)**



- Bit32-27: Reserved bits
Always set these bits to “00000”.
If a value other than “00000” is set, the operation of the MCU is not guaranteed.

- Bit26: Internal ROM enable
Specifies whether to enable the internal ROM area.

ROMA	Function	Remarks
0	External ROM mode	Enables the external ROM area.
1	Internal ROM mode	Enables the internal ROM area.

Always set to “1”.

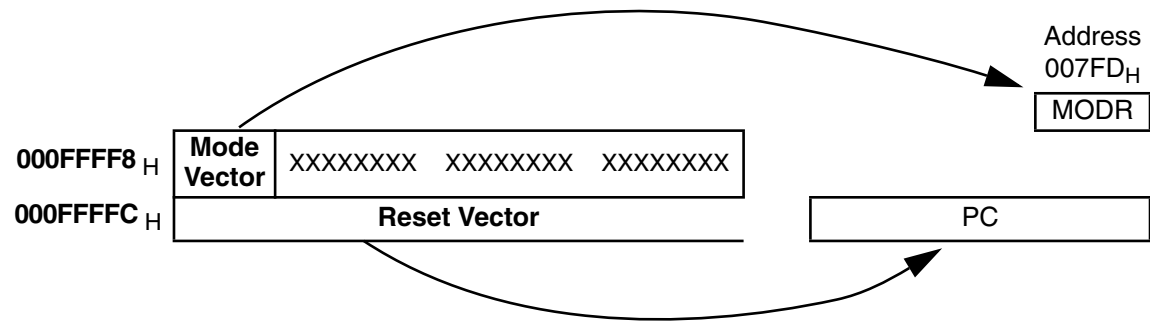
- Bit25-24: Bus width setting
This sets the bus width for external bus mode.

WTH1	WTH0	Function	Remarks
0	0	8-bit bus width	
0	1	16-bit bus width	
1	0	32-bit bus width	
1	1	Single chip mode	

- Bit23-0: Undefined bits

9.4.5 Reset Vector

The MCU starts program execution from the address specified by the reset vector.
Initial value to load into PC.



9.4.6 Device Mode Overview

The following table gives an overview about supported device mode combinations on the MB91460 series:

Mode pins			Mode/Reset Vector access area	FIXED Mode/Reset Vector	ROM A	ROM access area	WTH[1: 0]	Bus width	Remarks	
MD 2	MD 1	MD 0								
0	0	0	Internal	Yes	1	Internal	10	32bit	Fixed Mode Data is 0x06	
				No	0	External	00	8bit		
							01	16bit		
							10	32bit		
							11	Single	Setting not supported	
					1	Internal	00	8bit		
							01	16bit		
							10	32bit		
							11	Single		
0	0	1	External	-	0	External	00	8bit		
							01	16bit		
							10	32bit		
							11	Single	Setting not supported	
					1	Internal	00	8bit		
							01	16bit		
							10	32bit		
							11	Single	Setting not supported	
1	0	1	debug mode							
1	1	1	Flash test / parallel programmer mode							
others			Reserved							

- Remarks:
- On the MB91460 series the ROM area is from 0x0004:0000 up to 0xFFFF:FFFF

9.5. INITX Pin Input (INIT: Settings Initialization Reset)

9.5.1 Trigger

The INITX pin is used to trigger a settings initialization reset.

A settings initialization reset (INIT) request remains active while the pin remains at the “L” level. Keep the “L” level for the main oscillation stabilization time.

9.5.2 Releasing the Reset Request

Inputting an “H” level to the pin after the main oscillation stabilization time releases the settings initialization reset (INIT) request.

9.5.3 Flag

When an pin request triggers a settings initialization reset (INIT), the settings initialization reset flag (RSRR.INIT) is set to “1”.

9.5.4 Reset Level

This reset has the maximum reset level and initializes all settings. This type of reset is called the settings initialization reset (INIT)

A settings initialization reset (INIT) triggered by INITX pin input has the highest priority of all resets and has priority over all other inputs, operations, and states.

When a settings initialization reset (INIT) occurs, it is followed by an operation reset (RST) after the oscillation stabilization time elapses.

9.5.5 Initialization Triggered by INITX Pin Input (INIT)

- Device operation mode (bus mode and external bus width setting)
- All internal clock related settings (clock source selection, Main PLL control, division setting)
- All settings relating to the external bus CS0 area
- All other settings related to pin states
- All areas initialized by an operation reset (RST)
 - Program operation
 - CPU and internal bus
 - Peripheral circuit register contents
 - I/O port settings
- Device operation mode (bus mode and external bus width setting)

9.5.6 Reset Cancellation Sequence

After the cancellation (removal) of the settings initialization reset (external INITX pin) request the device performs the following operations in the sequence listed.

1. Removal of settings initialization reset (INIT)
2. Set operation reset (RST) state and start internal clock
3. Clear operation reset (RST) state and change to normal operation (RUN)
4. Read mode vector from address 000FFFF8_H
5. Write mode vector to MODR (mode register)
6. Read reset vector from address 000FFFFC_H
7. Write reset vector in PC (program counter)

Start program execution from the address specified by PC (program counter)

Note: See explanation of the time base counter at INITX input in section “[18.5. Operation \(Page No.369\)](#)”.

9.6. Hardware Watchdog / Clock Supervisor reset (INIT: Settings Initialization Reset)

9.6.1 Trigger

The Hardware Watchdog starts automatically after INIT is finished and cannot be disabled by software. The Clock Supervisor reset is enabled by default.

A Hardware Watchdog reset request is generated unless HWWDC.CL is cleared within the reset postponement register (HWWDE.ED[1:0]) period. The Clock Supervisor reset is generated when oscillation stops by fail.

9.6.2 Releasing the Reset Request

The Hardware Watchdog and Clock Supervisor reset request invokes a settings initialization reset (INIT). The watchdog reset request is released after the request is received and the settings initialization reset (INIT) generated.

9.6.3 Flags

When Hardware Watchdog reset request is triggered, the cpu reset flag (HWWDC.CPUF) is set to "1".

When the Clock Supervisor reset request is triggered, one of the flags CSVCR.MM or CSVCR.SM is set.

Additionally, both Hardware Watchdog and Clock Supervisor trigger a settings initialization reset (RSRR.INIT).

9.6.4 Reset Level

This reset has the maximum reset level and initializes all settings. This type of reset is called the settings initialization reset (INIT).

When a settings initialization reset (INIT) occurs, it is followed by an operation reset (RST) after the oscillation stabilization time elapses.

In case of a Clock Supervisor reset (the device is running on CLKRC then), it is necessary to have an external reset to start the device on CLKMAIN again. A Watchdog reset doesn't have this effect.

9.6.5 Initialization Triggered by Watchdog Reset (INIT)

Same as for a reset triggered by an INITX pin input.

However, the oscillation stabilization time selection bits (STCR.OS[1:0]) and reset cause flags (WDOG, SRST) are not initialized and retain their existing values.

In case of a Clock Supervisor reset (the device is running on CLKRC then), it is necessary to have an external reset or a power-on reset to start the device on CLKMAIN again. A Watchdog reset doesn't have this effect.

9.6.6 Reset Cancellation Sequence

Same as for INITX pin input.

(See "Chapter 21 Hardware Watchdog Timer (Page No.399)" as well as Chapter 16 Clock Supervisor (Page No.333) for details.)

9.7. Software Watchdog Reset (INIT: Settings Initialization Reset)

9.7.1 Trigger

Writing to the software watchdog timer control register (RSRR) starts the watchdog timer. Once started, a watchdog reset request is generated unless “A5_H” and “5A_H” are written to the watchdog reset delay register (WPR) within the time specified by the watchdog period selection bits (RSRR.WT[1:0]).

9.7.2 Releasing the Reset Request

The watchdog reset request invokes a settings initialization reset (INIT). The watchdog reset request is released after the request is received and the settings initialization reset (INIT) generated.

9.7.3 Flags

When Software Watchdog reset request is triggered, the watchdog timeout flag (RSRR.WDOG) is set to “1”.

9.7.4 Reset Level

This reset has the maximum reset level and initializes all settings. This type of reset is called the settings initialization reset (INIT).

When a settings initialization reset (INIT) occurs, it is followed by an operation reset (RST) after the oscillation stabilization time elapses.

9.7.5 Initialization Triggered by Watchdog Reset (INIT)

Same as for a reset triggered by an INITX pin input.

However, the oscillation stabilization time selection bits (STCR.OS[1:0]) and reset cause flags (INIT, SRST) are not initialized and retain their existing values.

9.7.6 Reset Cancellation Sequence

Same as for INITX pin input.

(See “[Chapter 20 Software Watchdog Timer \(Page No.389\)](#)” for details.)

9.8. Software Reset (RST: Operation Initialization Reset)

9.8.1 Trigger

Writing “0” to the software reset bit (STCR.SRST) generates a software reset request.
A software reset requests an operation reset (RST).

9.8.2 Releasing the Reset Request

The software reset request is released after the request is received and the operation reset (RST) generated.

9.8.3 Flag

When software reset request triggers an operation reset (RST), the software reset flag (RSRR.SRST) is set to “1”.

9.8.4 Reset Level

This is a normal level reset which only initializes the program and is called an operation reset (RST).
The following section lists the main items initialized by an operation reset (RST):

9.8.5 Items Initialized by Operation Reset (RST)

- Program operation
- CPU and internal bus
- Content of registers in peripheral circuits
- I/O port settings
- Device operation mode (bus mode and external bus width setting)

9.8.6 Reset Cancellation Sequence

After cancellation (removal) of the operation reset (RST) request, the device performs the following operations in the sequence listed.

1. Removal of operation reset (RST) and change to RUN state
2. Read mode vector from address 000FFFF8_H
3. Write mode vector to MODR (mode register)
4. Read reset vector from address 000FFFC_H
5. Write reset vector to the PC (program counter)
6. Start program execution from the address specified by the PC (program counter)

9.9. Reset Operation Modes

The following two different modes can be used for an operation reset (RST):

- Normal (asynchronous) reset mode
- Synchronous reset mode

Which mode to use is specified by the synchronous reset operation enable bit (TBCR.SYNCR).

Pin input resets and watchdog resets always use normal reset mode.

For software resets, either normal reset mode or synchronous reset mode can be selected.

9.9.1 Normal (Asynchronous) Reset Mode

Normal reset operation refers to the mode when the device goes to the operation reset (RST) state immediately after an operation reset (RST) request occurs.

For a normal reset, the device changes to the reset (RST) state immediately after a reset (RST) request is received regardless of the current state of internal bus access.

In normal reset mode, the result on any bus operation that is in progress at the time the device changes state is not guaranteed. However, acceptance of the operation reset (RST) request is guaranteed.

Setting the synchronous reset operation enable bit (TBCR.SYNCR) to "0" specifies normal reset mode.

Normal reset mode is the default setting after a settings initialization reset (INIT).

9.9.2 Synchronous Reset Operation

Synchronous reset operation refers to the mode when the device does not go to the operation reset (RST) state after a operation reset (RST) request until after all bus access has halted.

In synchronous reset mode, the device does not go to the reset (RST) state when a reset (RST) request is received if internal bus access is still in progress.

When such a reset request is received, a sleep request is issued to the internal bus. The device does not change to the operation reset (RST) state until all buses have shutdown operation and changed to SLEEP state.

In synchronous reset mode, the results of bus operations are guaranteed because the device does not change state until all bus access has halted.

However, if bus access should not halt for some reason, no requests can be received while bus operation continues. In such a case, the settings initialization reset (INIT) remains available at any time.

The following lists cases in which bus access may not stop:

If bus wait is enabled due to continuous input of RDY (ready request) to the external expansion bus interface.

MB91460 Series

9.10. MCU Operation Mode

After release of a reset, the MCU starts operation in the mode specified by the mode pins and mode data.

Operation mode	Bus mode	Single chip mode Internal ROM/external bus mode External ROM/external bus mode
	Access mode	32-bit bus width 16-bit bus width 8-bit bus width

9.10.1 Bus Modes and Access Modes

■ Bus mode

The bus mode controls internal ROM operation and the external access function. The bus mode is specified by the mode setting pins (MD2, MD1, MD0) and internal ROM enable bit (Mode-Vector. ROMA).

The FR-family CPU has the following three bus modes.

● Single chip mode

In this mode, internal I/O, internal RAM, and internal ROM are available but access to other areas is disabled. External pins are used either by the peripheral functions or as general-purpose ports. Pins cannot be used as bus pins. This mode can not be used when using the fixed mode/reset vector as implemented on most of the MB91460 series devices.

● Internal ROM, external bus mode

In this mode, internal I/O, internal RAM, and internal ROM are available, and access to areas for which external access is enabled results in access to the external area. Some external pins function as bus pins.

● External ROM, external bus mode

In this mode, internal I/O and internal RAM are available but access to internal ROM is prohibited. Access to internal ROM areas and areas for which external access is enabled results in access to the external area. Some external pins function as bus pins.

■ Access mode

The access mode controls the width of the external data bus and is set by the WTH[1:0] bits in the mode data.

9.11. Caution

- INITX pin input

Ensure that a settings initialization reset (INIT) is applied to this pin when the power is turned on. Also, after turning on the power, ensure a sufficient oscillation stabilization wait time is provided for the oscillation circuit by holding the input to the pin at the “L” level for the required time.

Note: The INIT reset triggered by INITX pin input initializes the oscillation stabilization wait time to its minimum value.

- Watchdog reset

When a settings initialization reset (INIT) is triggered by a watchdog reset request, the oscillation stabilization time is not initialized. Also, in Main-RUN or Sub-RUN state, no oscillation stabilization wait occurs in response to a watchdog reset request if the Main clock is not halted.

- Software reset

If “1” (synchronous reset mode) is set to the synchronous reset operation enable bit (TBCR.SYNCR) when an operation reset (RST) is triggered by a software reset request, the operation reset (RST) does not occur until all bus access halts. Accordingly, there may be a long delay before the operation reset (RST) occurs, depending on the bus usage.

- Settings initialization reset (INIT)

A settings initialization reset (INIT) invokes an operation reset (RST) after the oscillation stabilization wait time elapses.

- Reset cause flags (INIT, HSTB, WDOG, ERST, SRST and LINIT)

- Reading the reset cause register clears all the reset cause flags to “0”.
- If more than one reset occurs before the reset cause register is read, the flag values are combined (logical OR) and more than one flag may be set to “1”.

- Reset mode

A settings initialization reset (INIT) initializes the reset mode to normal reset mode.

- DMA controller

As the DMA controller halts any transfer when a request is received, it does not cause any delay in changing device state.

- Pin states during a reset

See “[3.10. Pin State Table \(Page No.192\)](#)” for details about pin states during a reset.

- Speciality for watchdog reset

If there happened a clock supervisor reset, MCU starts up on RC clock. If you now ran into a hardware watchdog reset by not triggering HWWD_CL bit in time, the bit CSVCR_MM (main clock missing) or CSVCR_SM (sub clock missing) is not cleared which leads to another startup on RC clock. Even if the hardware watchdog reset is an INIT reset type, it is not able to clear these bits.

To clear these bits and to startup on main clock again (only possible if previous clock supervisor reset was caused by a short-time distortion of main clock signal and not of a total loss of oscillator connection), you need power-on reset or external reset input at INITX pin.

Chapter 10 Standby

10.1. Overview

Three standby modes (low power consumption modes) are available.

- SLEEP state: Stops the program
- STOP state: Stops the device
- SHUTDOWN state: Shuts down the device

Note: It is possible to keep the Real Time Clock active in STOP / SHUTDOWN states (see chapter RTC).

10.2. Features

■ SLEEP state

- Device state in SLEEP state:
 - Halts the program.
 - CPU program execution only stops. Peripheral functions can continue to operate.
 - The internal memory and internal bus halt, until they are requested by DMA.
- Transition to SLEEP state:
 - SLEEP state is invoked by the program.
- Recovery from SLEEP state:
 - Generation of a valid interrupt request ends SLEEP state (returns to normal operation)
 - An INITX pin input invokes an initialization reset (INIT) followed by an operation reset (RST).

■ STOP state

- Device state in STOP state:
 - The overall device halts.
 - Internal circuits halt (with some exceptions)
 - Internal clock signals halt (with some exceptions)
 - Whether or not the oscillation circuit halts can be controlled by a setting (programmable).
 - All external pins can be set to high impedance (programmable, excludes some pins)
- Transition to STOP state:
 - STOP state is invoked by the program.
- Recovery from STOP state:
 - The following four interrupt requests change the device to the oscillation stabilization wait state.
 - External level-detect or edge-detect interrupt
 - Interrupt generated by oscillation stabilization wait timer for the Main clock when oscillation not halted.
 - Interrupt generated by oscillation stabilization wait timer for the Sub clock when oscillation not halted.
 - Real time clock interrupt when clk source to RTC is active.
 - Input to the INITX pin invokes an initialization reset (INIT) and then an operation reset (RST).

■ SHUTDOWN state (advanced STOP state)

- Device state in SHUT-DOWN state:
 - The big part of device shuts down (see [64.1. Overview \(Page No.1631\)](#)).
- Transition to STOP state:

- SHUT DOWN state is invoked by the program (see 64.4.1 Transition to shutdown state (Page No.1638)).
- Recovery from SHUT DOWN state:
 - The device recovers from SHUT DOWN state by several internal and external sources (see 64.4.2 Recovery from shutdown mode (Page No.1642)).

10.3. Configuration

Figure 10.3-1 Configuration Diagram of Standby operation

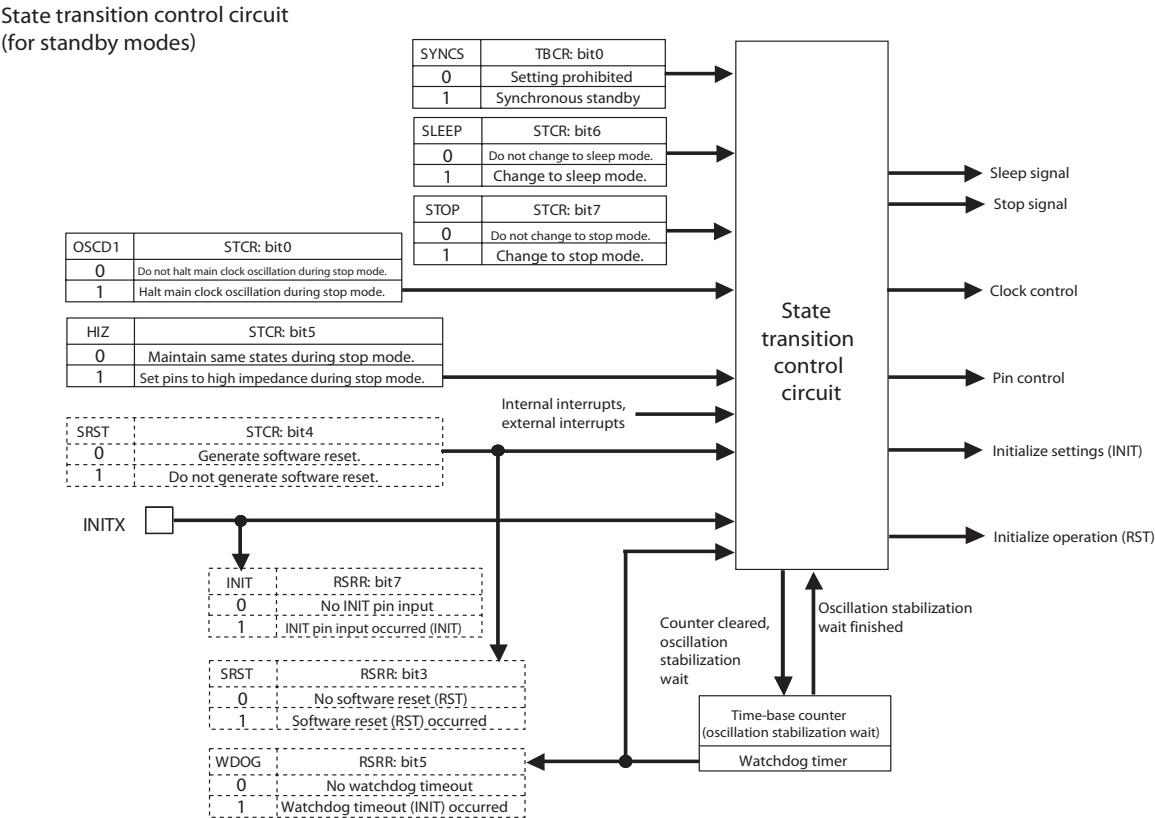


Figure 10.3-2 Register List

Standby Control

Address	Bit	7	6	5	4	3	2	1	0		
00480H		INIT	HSTB	WDOG	ERST	SRST	LINIT	WT1	WT0	RSRR	(Reset Cause)
00481H		STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	STCR	(Standby control)
00482H		TBIF	TBIE	TBC2	TBC1	TBC0	---	SYNCR	SYNCS	TBCR	(Time-base counter control)

MB91460 Series

10.4. Registers

10.4.1 STCR: Standby Control Register

Used to control transition to the STOP and SLEEP standby states, and to specify the pin states and whether to halt the oscillation during STOP state.

Note: See “[Chapter 9 Reset \(Page No.227\)](#)” also.

- **STCR: Address 0481h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	
0	0	1	1	0	0	1	1	Initial value (INITX pin input)
0	0	1	1	X	X	1	1	Initial value (Watchdog reset)
0	0	X	1	X	X	X	X	Initial value after RST (Software reset)
R/W	R/W	R/W	R1, W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7: STOP state

STOP	Operation
0	Does not change to STOP state.
1	Changes to STOP state.

- Goes to “0” when a reset (INITX pin input or software reset) occurs or on recovery from STOP state.
- When going directly from Main PLL operation to STOP state, the PLL operation should be disabled before the STOP state is invoked. (See “[10.8. Caution \(Page No.262\)](#)”.)

- Bit6: SLEEP state

SLEEP	Operation
0	Device does not change to SLEEP state.
1	Device changes to SLEEP state.

- If this bit and the STOP state bit (STOP) bit are set to “1” at the same time, the device goes to STOP state.
- Goes to “0” when a reset (INITX pin input or software reset) occurs or on recovery from SLEEP state.

- Bit5: High impedance mode

HIZ	Operation
0	Maintain same pin states when changing to STOP state.
1	Set pin outputs to high impedance (Hi-z) during STOP state.

- The default setting is high impedance.

- Bit4: Software reset (SRST)

- Setting this bit to “0” invokes a software reset.

- Bit3-2: Oscillation stabilization time selection (OS[1:0])

- Setting these bits in the range “00”-“11” sets the oscillation stabilization time to use after recovering from STOP state.

An INITX pin input reset or watchdog reset initialize this setting to its initial value.

(See “[Chapter 18 Timebase Counter \(Page No.363\)](#)”.)

- Bit1: Sub clock oscillation halt

OSCD2	Operation of Sub clock during STOP state
0	Continue oscillation
1	Halt oscillation

- Bit0: Main clock oscillation halt

OSCD1	Operation of Main clock during STOP state
0	Continue oscillation
1	Halt oscillation

10.4.2 TBCR: Timebase timer control register

This register controls the timebase timer interrupts and the options for resets and standby operation.

Note: See also “[Chapter 19 Timebase Timer \(Page No.379\)](#)”.

- **TBCR: Address 0482h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
TBIF	TBIE	TBC2	TBC1	TBC0	---	SYNCR	SYNCS	
0	0	X	X	X	X	0	0	Initial value (INITX pin, watchdog)
0	0	X	X	X	X	X	X	Initial value after RST (Software reset)
R(RM1), W	R/W	R/W	R1,W	R/W	RX/ WX	RX/WX	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7: Interrupt flag for timebase timer
 - This flag goes to “1” when a timebase timer interrupt occurs
- Bit6: Interrupt request enable for the timebase timer
 - Writing “1” to this bit enables timebase timer interrupt requests.
- Bit5-3: Interval time selection for timebase timer
 - Writing a value in the range “000”-“111” to these bits selects the interval time for the timebase timer.
($\phi \times 2^{11}$, $\phi \times 2^{12}$, $\phi \times 2^{13}$, $\phi \times 2^{22}$, $\phi \times 2^{23}$, $\phi \times 2^{24}$, $\phi \times 2^{25}$, $\phi \times 2^{26}$)
- Bit2: Reserved Writing does not affect the operation. The read value is undefined.
- Bit1: Enable synchronous reset operation
 - Selects a normal reset “0” or a synchronous reset “1”.
- Bit0: Enable synchronous standby operation

SYNCS	Operation
0	Asynchronous reset operation (Not permitted on this model).
1	Enable synchronous standby operation (always set this before changing to a standby mode).

MB91460 Series

10.5. Operation

10.5.1 SLEEP state

■ Entering SLEEP state

Writing “1” to the SLEEP state bit (STCR.SLEEP) changes to SLEEP state. The device remains in this mode until an event occurs to wakeup the device from SLEEP state.

(See “10.8. Caution (Page No.262)”.)

■ Device state in SLEEP state

- CPU program execution stops. (Peripheral functions continue to operate.)
- The internal memory and internal bus halt, until they are requested by DMA.
- Circuits that halt during SLEEP state
 - Bit search module
 - All internal memory (inclusive I-cache)
 - Internal/external bus
- Circuits that do not halt during SLEEP state
 - Oscillation circuit, Main PLL (if enabled)
 - Clock generation control circuit
 - Interrupt controller
 - External interrupts
 - DMA
 - Peripherals

■ Recovery and other items

- Generation of an interrupt request that is currently enabled changes the device back to RUN state. (Restores normal operation.)
- An INITX pin input invokes an initialization reset (INIT) followed by an operation reset (RST).

10.5.2 STOP state

■ Entering STOP state

Writing “1” to the STOP state bit (STCR.STOP) changes to STOP state.

The device remains in this mode until an event occurs to wakeup the device from STOP state.

(See “10.8. Caution (Page No.262)”.)

■ Device state in STOP state

- The overall device halts (internal circuits halt and the internal clock signals halt).
- Circuits that halt during STOP state
All internal circuits except those listed below.
- Circuits that do not halt during STOP state
 - Oscillation circuits that are not specified to be halted
 - Oscillation circuit for Main clock (if not disabled)
 - Oscillation circuit for Sub clock (if not disabled)
 - Main PLL circuit if oscillation circuit for Main clock is enabled and PLL circuit is enabled and main regulator is kept enabled. But the PLL circuit must be disabled before going to STOP state.
Please also refer to section 14.7.2 Main PLL Operation in STOP State (Page No.327).
 - Peripheral functions that are driven directly by the oscillation and which have not been specified to be halted.
 - Real Time Clock (if not disabled) and the RTC clock source is enabled.
 - LCDC (if LCD display enabled for Sub-STOP state and Sub clock selected as the clock source.)
- Pin states (High impedance or maintain previous state)
 - When pin outputs are set to go to high impedance during STOP state
 - High impedance output: Pins that are set as general purpose ports and pins that have been selected for use by peripheral functions.
 - When pin outputs are set to maintain their previous states during STOP state
 - Maintain previous state: Pins that are set as general purpose ports and pins that have been selected for use by peripheral functions.
 - When set as external interrupts
 - Input available state:
Pins set as external interrupt inputs using level detection or edge detection.
(Whether the pin output during STOP state has been set to either high impedance or maintain previous state.)

■ Recovery and other items

- Any of the following interrupt requests cause the device to go to the oscillation stabilization wait RUN state and then to change back to RUN state after the oscillation stabilization time elapses (return to normal operation).
 - External interrupts set to level detection or edge detection and that do not require a specific clock.
 - Real Time Clock interrupt (if operating)
- An INITX pin input or generation of a watchdog reset invokes an initialization reset (INIT) followed by an operation reset (RST) after the oscillation stabilization time.

MB91460 Series

10.6. Settings

Table 10.6-1 Settings Required to Change to SLEEP state

Setting	Setting register	Setting procedure*
Interrupt settings	(See the chapter for each peripheral function.)	—
Synchronous standby settings	Timebase timer control register (TBCR)	See 7.1
Change to SLEEP state	Standby control register (STCR)	See 7.1
Operational restrictions	(See “10.8. Caution (Page No.262)”.)	—

*:For the setting procedure, refer to the section indicated by the number.

Table 10.6-2 Settings Required to Change to STOP state

Setting	Setting register	Setting procedure*
Selects the oscillation stabilization wait time	(See “Chapter 18 Timebase Counter (Page No.363)”.)	—
Interrupt settings	(See the chapter for each peripheral function.)	—
Synchronous standby settings	Timebase timer control register (TBCR)	See 7.2
Change to STOP state	Standby control register (STCR)	See 7.2
Operational restrictions	(See “10.8. Caution (Page No.262)”.)	—

*: For the setting procedure, refer to the section indicated by the number.

10.7. Q&A

10.7.1 How to change to SLEEP state

Before a change to SLEEP state, first the synchronous standby operation enable bit (TBCR.SYNCS) must be set.

Operation	Synchronous standby operation enable bit (SYNCS)
To enable synchronous standby operation	Set to “1”.

Note: Setting (SYNCS=“0”) is prohibited.

Set using the SLEEP state bit (STCR.SLEEP).

Operation	SLEEP state bit (SLEEP)
Change not to SLEEP state	Set to “0”.
Change to SLEEP state	Set to “1”.

Note: Some restrictions apply when changing to SLEEP state. See “10.8. Caution (Page No.262)” for details.

10.7.2 How to change to STOP state

- When operating on the Main PLL clock, the operating clock must be set to the Main clock divided by two.
See “[13.7.3 How to select the operating clock source \(Page No.308\)](#)” for details about changing the operating clock.
- Before a change to STOP state, at first the synchronous standby operation enable bit (TBCR.SYNCS) must be set. See section 7.1.
- Set using the STOP state bit (STCR.STOP).

Operation	STOP state bit (STOP)
Change not to STOP state	Set to “0”.
Change to STOP state	Set to “1”.

Note: Some restrictions apply when changing to STOP state. See “[10.8. Caution \(Page No.262\)](#)” for details.

10.7.3 How to set pins to high impedance (Hi-z) during STOP state

Set using the high impedance mode bit (STCR.HIZ).

Operation	High impedance mode bit (HIZ)
Pins are not set to high impedance during STOP state	Set to “0”.
Pins are set to high impedance during STOP state	Set to “1”.

Note: Some ports do not go to high impedance in some circumstances. (See “[10.5.2 STOP state \(Page No.258\)](#)”.)

10.7.4 How to halt the Main clock oscillation during STOP state

Use the Main clock oscillation stop bit (STCR.OSCD1).

Operation	Main clock oscillation stop bit (OSCD1)
Main clock oscillation is not stopped during STOP state	Set to “0”.
Main clock oscillation is stopped during STOP state	Set to “1”.

10.7.5 How to recover from SLEEP state

Two methods are available to recover from SLEEP state.

- Generation of a valid interrupt request changes to RUN state (restores normal operation).
If using interrupt processing, remember to set the I flag (I), interrupt level mask register (ILM), and interrupt control register (ICR).
- An INITX pin input invokes an initialization reset (INIT) followed by an operation reset (RST).

MB91460 Series

10.7.6 How to recover from STOP state

The following events end STOP state:

- The following four interrupts change the device to the oscillation stabilization wait state.
 - External level-detect interrupt or edge-detect interrupt.
 - Oscillation stabilization wait timer for the Main clock when oscillation not halted.
 - Sub oscillation stabilization timer when oscillation not halted.
 - Real time clock when RTD clock source (Main Clock, Sub Clock or RC Oscillation) is active.

If using interrupt processing, remember to set the I flag (I), interrupt level mask register (ILM), and interrupt control register (ICR).

- Input to the INITX pin invokes an initialization reset (INIT) followed by an oscillation stabilization delay and then an operation reset (RST).

In the case of an INITX pin input, an oscillation stabilization wait is required, depending on the width of the INITX pin input.

See also “[Chapter 13 Clock Control \(Page No.289\)](#)” and “[Chapter 18 Timebase Counter \(Page No.363\)](#)”.

10.8. Caution

- Points to note when changing to SLEEP state

When changing to SLEEP state, set the synchronous standby operation enable bit (TBCR.SYNCS= “1”).

Also, in order to change to SLEEP state with synchronous standby operation enabled, the STCR register must be read after writing to the SLEEP bit. Always use the following sequence.

```
(LDI    #value_of_sleep, R0) ; value_of_sleep contains the write data for STCR.
(LDI    #_STCR, R12)         ; _STCR is the address of STCR (481H).
STB     R0, @R12             ; Write to standby control register (STCR).
LDUB    @R12, R0             ; STCR read required for synchronous standby.
LDUB    @R12, R0             ; Second dummy read to STCR.
NOP     }                    ; NOP x 5 required (for flushing the pipeline)
NOP     }
NOP     }
NOP     }
NOP     }
```

- Points to note when changing to STOP state

When changing to SLEEP state, set the synchronous standby operation enable bit (TBCR.SYNCS= “1”).

Also, in order to change to STOP state with synchronous standby operation enabled, the STCR register must be read after writing to the STOP bit. Always use the following sequence.

```
(LDI    #value_of_stop, R0) ; value_of_stop contains the write data for STCR.
(LDI    #_STCR, R12)         ; _STCR is the address of STCR (481H).
STB     R0, @R12             ; Write to standby control register (STCR).
LDUB    @R12, R0             ; STCR read required for synchronous standby.
LDUB    @R12, R0             ; Second dummy read to STCR.
NOP     }                    ; NOP x 5 required (for flushing the pipeline)
NOP     }
NOP     }
NOP     }
NOP     }
```

- When the Main PLL is selected as the operation clock source

When the Main PLL is selected as the operation clock source, change the operation clock source selection to Main clock divided by two before changing to STOP state.

See “[Chapter 13 Clock Control \(Page No.289\)](#)” for details.

Please also refer to section [14.7.2 Main PLL Operation in STOP State \(Page No.327\)](#).

The restrictions that apply to the clock divide ratio setting are the same as for normal operation. The PLL oscillation has also not to be halted.

- If interrupts are disabled in the interrupt control register (ICR=“0001111B”), the device will not recover from STOP or SLEEP state when an interrupt occurs.
- Pin high impedance control in STOP state

Setting the high impedance bit (STCR.HIZ) to “1” sets pin outputs to high impedance during STOP state. If the high impedance bit (STCR.HIZ) is set to “0”, pins retain the states they have prior to entering STOP state.

See [Chapter 3.8 "Pin State Table" on P. 192](#) for details such as the operation of specific pins

Chapter 11 Memory Controller, Flash and F-Cache

11.1. Overview

This module combines the interfaces to the F-Bus memory resources, FLASH and General Purpose RAM (also referenced as ID-RAM).

These memories can be combined CODE and DATA storage. While code fetch is possible in general via the F-Bus at the FR core, due to performance reasons the code fetch is accelerated by a direct I-Bus connection in MB91460 series MCUs.

For FLASH access the interface contains an instruction cache (called Flash Cache or F-Cache) and data read buffer. A prefetch mechanism removes CPU internal code fetch latencies for linear code.

In addition the module includes the definition of the Fixed Mode Vector (FMV) and the Fixed Reset Vector (FRV), depending on the device mode.

11.2. Main Flash Interface

- Wait timing
- Generation of FLASH control signals ATDIN and EQIN for synchronous access.
(this version supports independent timing configuration of ADTIN, EQIN and Wait)
- Generation of CEX, WEX and OEX
- Handling of 32 or 64 bit read mode and 16 or 32 bit read/write mode for programming
- Support of external SRAM for emulation devices with 1:1 timing transparency (same wait cycles)
- Measures for FLASH macro test and parallel programming support

11.3. Embedded ID-RAM

- Zero wait cycle access (code), one wait cycle access (data) to shared code/data memory (up to 64 kByte), also referenced as ID-RAM

11.4. Flash Cache and Data Buffer

- Up to 16 kByte Instruction cache (4k word entries, one way direct mapped, prefetch miss option)
- Size configuration for the evaluation device (0, 4, 8 and 16 kB)
- 1 or 2 dword (32 or 64 bit) data read buffer (**64 bit not available on MB91V460A**)

11.5. Prefetch

- Prefetch of consecutive instruction word address to the cache buffer
- Prefetch is canceled in case of prefetch miss (branch or data access), thus it works without any penalties in the prefetch miss case.
- The FLASH macro needs to support FLASH access cycle cancelation at any point, that means it may not affect the timing of the next complete access cycle (no special recovery condition required from previous access cancelation).

11.6. Fixed Mode and Reset Vectors

- Mode vector address: 0x000ffff8; return 0x06000000 for internal vector mode
- Reset vector address: 0x000fffc; return 0x00030000 at RAM execution mode (jump to test program) or return 0x0000bff8 in any other case (jump to Boot ROM)
- If FMCS_FIXE is switched off, the FLASH memory can be accessed on addresses 0x000ffff8 and 0x000fffc. FIXE is set at reset.

11.7. Registers

- List of FLASH-Interface (FLASH-IF) Registers

Table 11.7-1 FLASH-IF Registers Summary

Address	Register				Block
	+0	+1	+2	+3	
7000 _H	FMCS [R/W] 01101000	FMCR [R/W] - ---- 0000	FCHCR [R/W] -----00 1000011		FLASH_IF
7004 _H	FMWT[R/W] 11111111 01111111 ¹ 11111111 01011101 ²		FMWT2[R/W] -101----	FMPS[R/W] ---- 000	
7008 _H	FMAC [R] 00000000 00000000 00000000 00000000				
700C _H	FCHA0 ----- -0000000 00000000 00000000				
7010 _H	FCHA1 ----- -0000000 00000000 00000000				

1. Initial value on MB91V460A
2. Initial value on MB91FV460B and all devices with embedded flash memory

Remarks:

- Read and write access to all registers is byte, halfword and word.
- FMCR and FMWT2 registers are not available on MB91V460A
- The initial values of the wait cycle setting register FMWT differs between the flash-less evaluation device MB91V460A and devices with embedded flash memory

11.8. Explanations of Registers

● FLASH Interface Control Register

Control Register byte 0	31	30	29	28	27	26	25	24	← Bit no.
Address : 7000H	ASYNC	FIXE	(BIRE)	RDYEG	RDY	RDYI	RW16	-	FMCS
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R)	(R)	(R/W)	(R/W)	(-)	
Default value⇒	(0)	(1)	(1)	(0)	(1)	(0)	(0)	(X)	

Control Register byte 1	23	22	21	20	19	18	17	16	← Bit no.
Address : 7001H	-	-	-	-	LOCK	PHASE	PF2I	RD64	FMCR
Read/write ⇒	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(X)	(X)	(X)	(X)	(0)	(0)	(0)	(0)	

Control Register byte 2	15	14	13	12	11	10	9	8	← Bit no.
Address : 7002H	-	-	-	-	-	-	REN	(TAGE)	FCHCR
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(X)	(X)	(X)	(X)	(X)	(X)	(0)	(0)	

Control Register byte 3	7	6	5	4	3	2	1	0	← Bit no.
Address : 7003H	FLUSH	(DBEN)	PFEN	PFMC	LOCK	ENAB	SIZE1	SIZE0	FCHCR
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(0)	(0)	(0)	(0)	(0)	(1)	(1)	

11.8.1 FLASH Memory Control and Status Register (FMCS)

• BIT[31]: ASYNC - ASYNChronous access enable

0	Synchronous FLASH access (default)
1	Asynchronous FLASH access

The ASYNC bit is cleared at reset, which enables the fast synchronous FLASH access mode by default. To switch to asynchronous mode, set this bit (however it is basically not recommended to set this bit, neither in read nor in write access).

• BIT[30]: FIXE - FIXed reset and mode vector Enable

0	Disable FMV/FRV and enable FLASH access at mode vector address
1	Output the fixed mode or reset vector at address hit (default)

The FIXE bit is set by default.

To enable FLASH access on address 0x000ffff8 and 0x000ffffc, clear this bit.

- **BIT[29]: BIRE - Burn-In ROM Enable**

0	Disable Burn-In ROM and enable FLASH access at Burn-In ROM address
1	Enable access to the Burn-In ROM (default)

The BIRE bit is a reserved bit and should not be used.

- **BIT[28]: RDYEG - RDY status hold register**

0	Auto Program Algorithm not started or started and not completed (default)
1	The FLASH Auto Program Algorithm has been completed since last register read access.

The RDYEG bit is cleared after reset.

The bit is set after 0->1 transition of FMCS_RDY. The bit shows that the RDY signal of the FLASH is or was active (completed Auto Program Algorithm). The RDYEG bit is cleared automatically at read access to address 0x7000.

The RDYEG bit is read-only status information.

Remark: The function of this status bit is not guaranteed when running the CPU on frequencies lower than 1MHz.

- **BIT[27]: RDY - FLASH RDY status of Auto Program Algorithm**

This bit shows the status of the RDY line of the FLASH macro. RDY is used to signalize the state of the FLASH macro in case of an Auto Program Algorithm was started (e.g. sector erase, chip erase). If RDY returns to '1', the Auto Program Algorithm has been completed.

The RDY bit is read-only status information.

Remark: The function of this status bit is not guaranteed when running the CPU on frequencies lower than 1MHz.

- **BIT[26]: RDYI - RDY output force**

0	Inactive (default)
1	Force the RDY output to '1'

This bit is reserved for FLASH test. Do not set this bit.

- **BIT[25]: RW16 - 16 bit Read/Write enable to FLASH**

0	32 bit read and write access to FLASH is enabled (default)
1	16 bit read and write access to FLASH is enabled

This bit is cleared after reset. There is a 32 bit read and write access to the FLASH memory enabled by default.

Setting of the RW16 bit implies switching from 32 bit into 16 bit mode. When it is intended to write data to the flash memory (or while chip erase or sector erase) then code fetch from the flash memory is not supported.

Important remark: To maintain data consistency it is strongly recommended to disable the F-Cache while writing to the FLASH memory and to flush the F-Cache (FLUSH=1) after completing the write procedure to the FLASH memory.

Important remark: It is not allowed to switch between the 16 bit, the 32 bit and the 64 bit mode while reading instructions or data from the FLASH memory. Please refer to section 54.6. [Flash Access Mode Switching \(Page No.1183\)](#).

- **BIT[24] - BIT[20]: Reserved**

11.8.2 FLASH Memory Control Register (FMCR)

The FMCR register is not available on the evaluation device MB91V460A.

- **BIT[19]: LOCK - ALEH auto-update lock**

0	ALEH setting auto update is enabled (default)
1	ALEH setting auto update is disabled

FLASH memories embedded on the MB91460 series require a certain timing between ATDIN falling edge and EQIN rising edge. This timing is named tALEH and has usually the same length as the ATDIN duration.

By writing the setting of ATDIN length to the FMWT.ATD[2:0] bits, the FMWT2.ALEH[2:0] bits will be updated automatically to the same setting. To avoid this automatic update it is possible to set the ALEH LOCK bit.

It is also possible to apply a different setting to the FMWT2.ALEH[2:0] bits by writing first to the FMWT.ATD[2:0] bits and second to the FMWT2.ALEH[2:0] bits.

- **BIT[18]: PHASE - ATDIN/EQIN clock phase**

0	ATDIN/EQIN generation is in phase with the core clock (default)
1	ATDIN/EQIN generation is inverted to the core clock

At lower core clock frequencies it can be beneficial to change the ATDIN/EQIN generation to inverted core clock to save a wait cycle compared to the generation of these signals in phase with the core clock.

It is recommended to always refer to the setting requirements of ATDIN, EQIN and wait cycles for each product which are provided by Fujitsu (see the related datasheets).

Remark: PHASE setting is not available on MB91460 series.

- **BIT[17]: PF2I - Prefetch 32 bit (2 instructions) only**

0	Prefetch 64 bit (default)
1	Prefetch 32 bit only

When switching on 64 bit read mode (RD64=1) then prefetch will be performed on instruction address IA+8 (when current access is aligned at IA+0) and on instruction address IA+4 (when current access is aligned at IA+4). However, the setting of PF2I=1 in the 64 bit read mode will cause a prefetch only on next instruction address IA+4 (independent of current access alignment is IA+0 or IA+4).

Usually prefetching 64 bit is superior to 32 bit only, however it can be the case on strong fragmented code that

the performance deteriorates due to replacement of cache entries. In this case it can be sensible to switch to 32 bit prefetch only.

- **BIT[16]: RD64 - Enable 64 bit read mode**

0	64 bit read mode is disabled (default)
1	64 bit read mode is enabled

Some embedded FLASH memories support switching the 64 bit read mode to increase the access performance. Please refer to [Table 54.6-1 MB91460 series CPU flash access modes \(Page No.1183\)](#) to check the availability of the 64-bit read mode.

This bit is cleared after reset. The 32 bit read and write access to the FLASH memory is enabled by default.

Setting of the RD64 bit implies switching from 32 bit into 64 bit mode. Writing data to the flash memory is not supported in the 64 bit read only mode.

Important remark: It is not allowed to switch between the 16 bit, the 32 bit and the 64 bit mode while reading instructions or data from the FLASH memory. Please refer to section [54.6. Flash Access Mode Switching \(Page No.1183\)](#).

11.8.3 FLASH Cache Control Register (FCHCR)

- **BIT[9]: REN - Non-cacheable area Range Enable**

0	FCHA1 defines address mask (default)
1	FCHA1 defines second point for the non-cacheable address range from FCHA0 to FCHA1

The bit is cleared after reset. The address defined in FCHA0 is combined with a bit mask defined in FCHA1 to define the non-cacheable area.

If the REN bit is set, the non-cacheable area is defined by two points. The non-cacheable range is from addresses greater than or equal to FCHA0 up to addresses less than or equal to FCHA1.

- **BIT[8]: TAGE - TAG RAM access Enable**

0	Memory mapped TAG RAM access disabled (default)
1	Memory mapped TAG RAM access enabled

The bit is set to 0 after reset.

Remark: TAG RAM access is not available on MB91460 series.

- **BIT[7]: FLUSH - Flush instruction cache entries**

0	Flushing the instruction cache entries has been completed
1	Actually flushing the instruction cache entries

This bit is set after reset.

If the FLUSH bit is set, the instruction cache entries are flushed sequentially. During this initialization the cache is disabled. The initialization has a duration of 1 clock cycle per cache entry. The number of valid entries depends on the configured cache size.

After completion (all entries are flushed) the FLUSH bit is cleared by hardware.

Writing a '1' to this bit triggers the flushing of the cache entries.

Important remark: It is not allowed to set the cache size configuration (FCHCR.SZ[1:0]) and FLUSH at the same time (same write access). Always first set the size configuration before flushing the cache.

• **BIT[6]: DBEN - Data Buffer Enable**

0	Buffering of read data is disabled (default)
1	Buffering of read data is enabled

This bit is cleared after reset. The read data buffer is disabled by default.

Setting the DBEN bit enables the data read buffer. This is useful to speed up reading of data structures of 8 or 16 bit operands. There is one word data buffer implemented. If the same 32 or 64 bit word address is accessed consecutively, the data is read from the buffer.

Remark: Data buffer is not available on MB91V460A.

• **BIT[5]: PFEN - PreFetch ENable**

0	Prefetch of instructions is disabled (default)
1	Prefetch of instructions is enabled

This bit is cleared after reset. The prefetch of instructions is disabled by default.

Setting the PFEN bit enables the code prefetch from the next word on instruction address IA+4. Prefetch eliminates any latency in the code fetch path of the MCU to the FLASH memory for linear code.

When switching on 64 bit read mode (RD64=1) then prefetch will be performed on instruction address IA+8 (when current access is aligned at IA+0) and on instruction address IA+4 (when current access is aligned at IA+4). However, the setting of PF2I=1 in the 64 bit read mode will cause a prefetch only on next instruction address IA+4 (independent of current access alignment is IA+0 or IA+4).

A running prefetch cycle can be directly taken over from a matching instruction access. If there is no instruction access in between, the prefetched instruction word is stored in cache memory. If there is an FLASH access (code or data) to an address different from the prefetch address, the prefetch cycle is canceled immediately.

• **BIT[4]: PFMC - Prefetch Miss Cache enable**

0	Standard cache algorithm (default)
1	Prefetch misses are cached only

This bit is cleared after reset. The prefetch miss cache is disabled by default. The instruction cache uses the standard algorithm of writing cache entries for each accessed instruction word from FLASH.

Setting the PFMC bit switches to a second write algorithm for cache entries. This algorithm writes only this instruction words to the cache, which are causing prefetch miss conditions.

The FR CPU requests approximately one instruction word (which contains two 16 bit instruction codes) in two clock cycles. If the FLASH data throughput (one word in two cycles) is sufficient for the needs of the CPU, the PFMC option is useful in most cases.

If the FLASH access time is two clock cycles, normally no wait states are generated when the next instruction word is requested from a consecutive address and prefetch is enabled. Thus, caching such linear code segments in conjunction with prefetch may not improve the code fetch performance, which is at the optimum already. More interesting is to improve the situation for branches in the code, where prefetch could not remove the latency of accessing it. If FPMC is set to '1', the cache algorithm stores only these FLASH accesses, which have caused a wait condition due to a prefetch miss condition (not matched predicted address).

The effect of this algorithm is, that the restricted amount of cache entries is utilized more efficiently. Usually the same performance can be reached with half the cache size. Or, in other words, the cache is as same efficient as it would have the doubled size.

The efficiency of the PFMC algorithm depends on the structure of the application.

- **BIT[3]: LOCK - Global lock of cache entries**

0	Write of cache entries enabled (default)
1	Writing of cache entries is disabled, the cache contents is locked

This bit is cleared during reset. The cache entries are writable by default.

If the LOCK bit is set, no new entries can be written to cache memory. The old contents of cache entries remains in memory. There is only a global lock feature for all cache entries.

- **BIT[2]: ENAB - Instruction cache enable**

0	The instruction cache is disabled (default)
1	Enable the instruction cache

This bit is cleared after reset. By default the instruction cache is disabled.

If the ENAB bit is set, the instruction cache is switched on. The instruction cache is dedicated to FLASH access only.

The cache is utilized by the prefetch algorithm as prefetch buffer. Hence prefetch can be used in an unbuffered form with cache disabled.

Cache miss did not cause code fetch penalties. The FLASH access is started in parallel, independent from cache hit or miss evaluation.

(If the cache is disabled, the cache entries and the TAG RAM contents can be accessed memory mapped. This feature is disabled in this version of the interface, see the explanation of the TAGE bit.)

- **BIT[1:0]: SZ[1:0] - Cache size configuration**

00	0kByte - Cache disabled
01	4kByte (1024 entries)
10	8kByte (2048 entries)
11	16kByte (4096 entries) (default)

The cache size is set to '11' after reset.

Remark: The number of cache entries determines the TAG initialization period at device startup, see the explanation of the FLUSH bit above.

Important remark: On products with less than 16kByte instruction cache it is recommended to set the size configuration in the system startup according to the available cache size.

Important remark: It is not allowed to set the cache size configuration (FCHCR.SZ[1:0]) and FLUSH at the same time (same write access). Always first set the size configuration before flushing the cache.

11.8.4 FLASH Memory Wait Timing Register (FMWT)

Wait Timing Register byte 0	31	30	29	28	27	26	25	24	← Bit no.
Address : 7004H	WTP1	WTP0	WEXH1	WEXH0	WTC3	WTC2	WTC1	WTC0	FMWT
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Wait Timing Register byte 1	23	22	21	20	19	18	17	16	← Bit no.
Address : 7005H	(FRAM)	ATD2	ATD1	ATD0	EQ3	EQ2	EQ1	EQ0	FMWT
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1/0)	(1)	(1)	(1)	(1/0)	(1)	

Wait Timing Register byte 2	15	14	13	12	11	10	9	8	← Bit no.
Address : 7006H	-	ALEH2	ALEH1	ALEH0	-	-	-	-	FMWT2
Read/write ⇒	(-)	(1)	(0)	(1)	(-)	(-)	(-)	(-)	
Default value⇒	(X)	(R/W)	(R/W)	(R/W)	(X)	(X)	(X)	(X)	

Wait Timing Register byte 3	7	6	5	4	3	2	1	0	← Bit no.
Address : 7007H	-	-	-	-	-	PS2	PS1	PS0	FMPS
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	
Default value⇒	(X)	(X)	(X)	(X)	(X)	(0)	(0)	(0)	

Remarks:

- ATD[2:0] setting is 0x7 on MB91V460A and 0x5 on MB91FV460B and all other flash devices
- EQ[3:0] setting is 0xF on MB91V460A and 0xD on MB91FV460B and all other flash devices
- FMWT2 is not available on MB91V460A

- **BIT[31:30]: WTP[1:0] - Wait cycles for FLASH in page access**

WTP is set to 3 after reset.

WTP controls the wait timing of the FLASH access in case of page hit for Page Mode FLASH. The WTP configuration is in units of clock cycles. The value of WTP should be set to the intra page access time (cycle time) of the FLASH memory in number of clock cycles, subtracted by one.

The setting is used if the page size PS[2:0] is set different to 0.

- **BIT[29:28]: WEXH[1:0] - Minimum WEX High timing requirement**

WEXH is set to 3 after reset. The minimum high time duration of WEX is 5 cycles by default.

Setting an other value reduces the WEX high time to 2 fixed cycles + WEXH.

• **BIT[27:24]: WTC[3:0] - Wait cycles for FLASH memory access**

WTC is set to 15 after reset.

WTC controls the wait timing of the FLASH access. The WTC configuration is in units of clock cycles. The value of WTC should be set to the access time (cycle time) of the FLASH memory in number of clock cycles, subtracted by one.

• **BIT[23]: FRAM - Wait cycles for F-Bus general purpose RAM memory access**

FRAM is set to 0 after reset.

This is a reserved bit. This version on MB91V460A has no configurable wait timing to F-Bus RAM, it operates with fixed 0 wait states RAM access.

• **BIT[22:20]: ATD[2:0] - Duration of the ATDIN signal for FLASH memory access**

MB91V460A: ATD is set to 7 after reset. ATD defaults to 4 clock cycles.

MB91FV460B and all other flash devices: ATD is set to 5 after reset. ATD defaults to 3 clock cycles.

ATD controls the timing of the ATDIN signal for FLASH access. The ATD configuration is in units of half clock cycles. The effective high duration of ATDIN equals to $t_{\text{ATDIN}} = (\text{ATD} + 1) * 0.5$ clock cycles.

• **BIT[19:16]: EQ[3:0] - Duration of the EQIN signal for FLASH memory access**

MB91V460A: EQ is set to 15 after reset. EQ defaults to 8 clock cycles.

MB91FV460B and all other flash devices: EQ is set to 13 after reset. EQ defaults to 7 clock cycles.

EQ controls the timing of the EQIN signal for FLASH access. The EQ configuration is in units of half clock cycles. The effective high duration of EQIN equals to $t_{\text{EQIN}} = (\text{EQ} + 1) * 0.5$ clock cycles.

• **BIT[14:12]: ALEH[2:0] - Duration of the ALEH time for FLASH memory access**

MB91V460A: not available

MB91FV460B and all other flash devices: ALEH is set to 5 after reset. ALEH defaults to 3 clock cycles.

ALEH controls the timing of the ATDIN falling edge to EQIN rising edge for FLASH access.

The EQ configuration is in units of half clock cycles. The effective duration of ALEH equals to $t_{\text{ALEH}} = (\text{ALEH} + 1) * 0.5$ clock cycles.

Important remark: ALEH[2:0] is updated automatically to the same value as ATD[2:0] when writing to ATD[2:0]. Usually the ALEH time equals the ATD time, so there is normally no reason to update ALEH[2:0] in particular.

Even though it is possible to program ALEH[2:0] with a different value than ATD[2:0] by:

- Writing a different value to ALEH[2:0] after writing to ATD[2:0], or
- Setting the FMCR.LOCK bit to disable the auto update

11.8.5 FLASH access cycle waveform

Figure 11.8-1 Timing of a FLASH access cycle

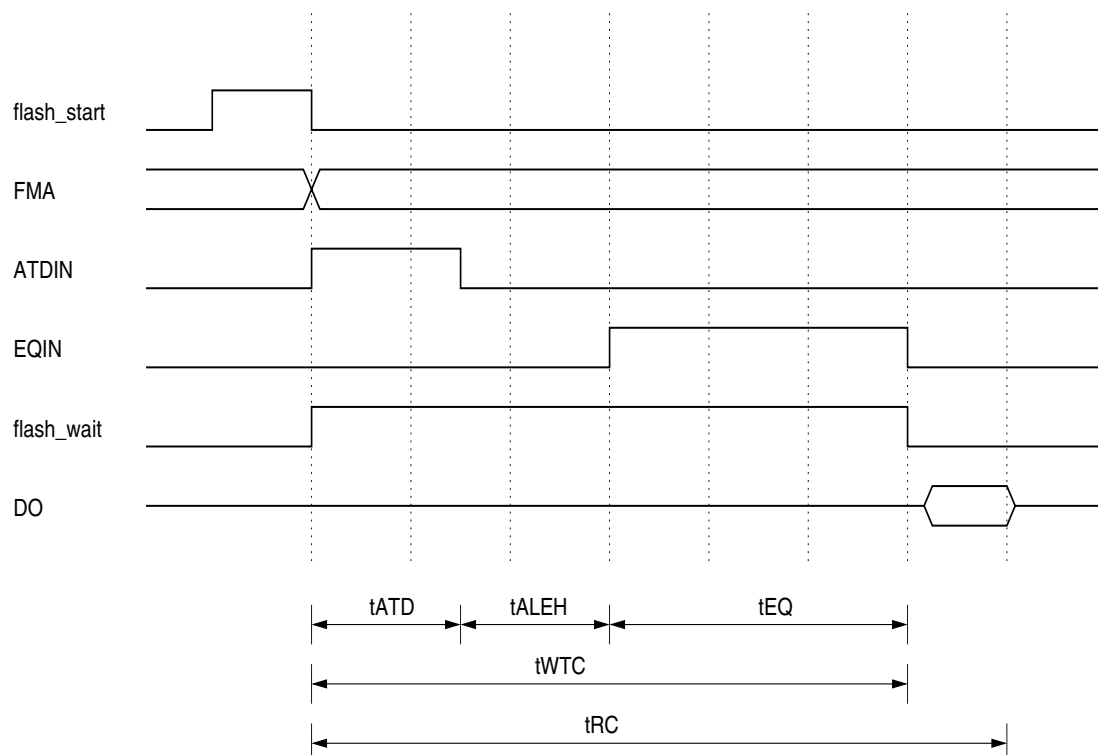


Figure 11.8-1 shows the example of a FLASH access cycle. In the FMWT register the three parts of the FLASH timing tATD, tALEH, tEQ and tWTC can be configured independently. The table below lists the configuration values for this example.

Symbol	Length	Setup
tATD	1.5 cycles	ATD=2
tALEH	1.5 cycles	ALEH=2
tEQ	3 cycles	EQ=5
tWTC	6 cycles	WTC=6

The resulting FLASH access cycle (tRC) time is 7 cycles (WTC+1).

- **BIT[2:0]: PS[2:0] - Page size definition for Page Mode FLASH**

PS is set to 0 after reset. Page Mode FLASH is disabled by default.

This setting defines the page size to 2^{PS} in number of bytes.

E.g. for Am29PL320D/MBM29PL3200 with a page size of 16 byte the value of PS has to be set to 4.

Remark: Embedded FLASH memories on MB91460 series do not support page mode.

11.8.6 FLASH Memory Address Check register (FMAC)

FMAC [R]				
Address	+0	+1	+2	+3
7008 _H	-----	-0000000	00000000	00000000

This register captures the address at the begin of a FLASH access cycle for test purposes.

The register could be read only.

11.8.7 Non-cacheable area definition registers (FCHA0, FCHA1)

The non-cacheable area definition registers FCHA0 and FCHA1 define the FLASH region not to be cached. Not used bits are read back as zero. The tables below define the initial values of the registers. The point defined by FCHA0 = 0 and mask bits off is located outside the FLASH region, thus initially the whole FLASH region will be cached.

FCHA0 [R/W]				
Address	+0	+1	+2	+3
700C _H	-----	-0000000	00000000	00000000

FCHA1 [R/W]				
Address	+0	+1	+2	+3
7010 _H	-----	-0000000	00000000	00000000

If the FCHCR_REN bit is cleared, the address range is defined by the address given by FCHA0, masked with the bits set to '1' in FCHA1.

Example 1 (Point and mask range definition):

- FCHCR_REN = 0

- FCHA0 = 0x000F:A300

- FCHA1 = 0x0000:FFFF

The non-cacheable area is defined from 0x000F:0000 to 0x000F:FFFF.

Example 2 (Point to point range definition):

- FCHCR_REN = 1

- FCHA0 = 0x000F:A300

- FCHA1 = 0x000F:F7FF

The non-cacheable area is defined from 0x000F:A300 to 0x000F:F7FF.

Chapter 12 External Bus Instruction Cache (I-Cache)

This chapter describes the instruction cache (I-Cache) memory for the External Bus included in MB91460 family members and its operation.

Please check section [2.3.2 MB91460 Series Device Lineup \(Page No.27\)](#) to find out which MB91460 series devices are equipped with the I-Cache.

12.1. Overview

The instruction cache is a temporary memory. When an instruction code is accessed from an external low speed memory, the cache retains internally the code which has been accessed once, and the code is used to make the second and later access faster.

When the instruction cache is enabled and then prohibited, always use the subroutine described in section [12.7. Setting Method When I-cache is Used \(Page No.286\)](#).

The devices have 4 KB RAM for the instruction cache. The area to be allocated for the instruction cache can be selected from 4 KB, 2 KB, and 1 KB.

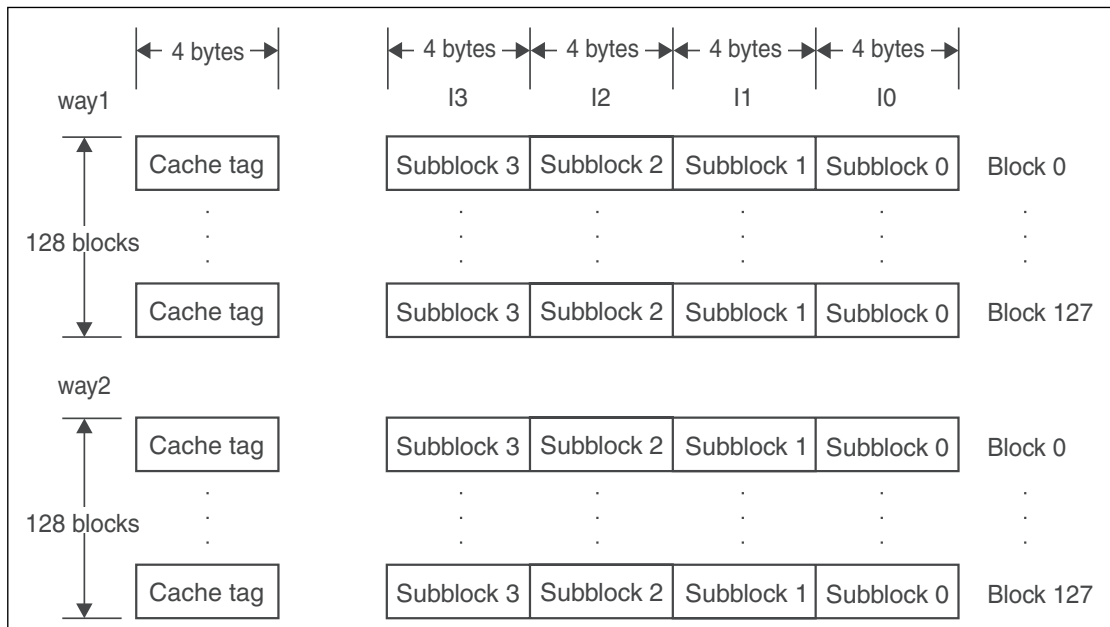
The remaining RAM is mapped on the memory as I-RAM.

When the instruction cache data RAM and TAG-RAM¹ is set to the RAM mode, it can be read/write-accessed directly with software.

12.2. Main Unit Structure

- FR basic instruction length: 2 bytes
- Block arrangement system: 2-way set associative system
- Block: 128 blocks per way
16 bytes per block (= 4 sub-blocks)
4 bytes per sub-block (= 1 bus access unit)

Figure 12.2-1 Instruction Cache Structure

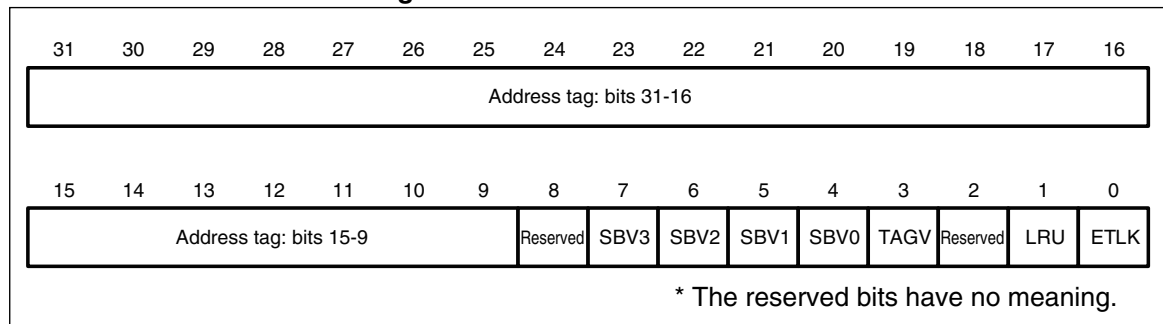


1. For the TAG-RAM access, there are restrictions. Not all bits can be used.

12.3. Tag Structure

The tag structure of the instruction cache is shown in [Figure 12.3-1](#).

Figure 12.3-1 Instruction Cache Tag Structure



- [Bits 31 to 9] **Address tag**

This area stores the upper 23 bits of the memory address of the instruction cached in the corresponding block. For example, memory address **IA** of the instruction data stored in sub-block **k** in block **i** is obtained from the following equation:

$$IA = \text{address tag} \times 2^{11} + i \times 2^4 + k \times 2^2$$

The address tag is used to test the agreement to the instruction address required with the access from CPU. The following behavior is performed according to the tag test result.

- 1) When the instruction data required exists in the cache (hit), the cache transfers the data to CPU within the cycle.
- 2) When the instruction data required does not exist in the cache (mistake), CPU and the cache simultaneously acquire the data acquired with an external access.

- [Bit 8] **Reserved**: This bit always reads '0'

- [Bits 7 to 4] **SBV3 to SBV0**: Sub-block valid bits

When SBV* contains "1", the current instruction data for the address indicated with a tag in the corresponding subblock is entered. Normally 2 instructions are stored in the subblock (immediate value transfer instruction excluded).

- [Bit 3] **TAGV**: **TAG** valid bit

This bit indicates whether the address tag value is valid. When the bit contains "0", the corresponding block is invalid regardless of the settings of the sub-block validation bits. (The FLUSHbit is set to "0" when the cache is flushed.)

- [Bit 2] **Reserved**: This bit always reads '0'.

- [Bit 1] **LRU** bit (way 1 only)

This bit exists only in the instruction cache tag in way 1. The bit indicates way 1 or 2 as the way containing the last entry accessed in the selected set. When set to "1", the LRU bit indicates that the entry of the set in way 1 is the last entry accessed. When set to "0", it indicates that the one in way 2 is the last entry accessed.

- [Bit 1] **Reserved** (way 2 only): This bit always reads '0'.

- [Bit 0] **ETLK**: Entry lock bit

All subblocks entries in the tag are locked.

When ETLK="1", entries are in locked status, and they are not updated on cache mistake.

However, an invalid subblock is updated.

When both way1 and way2 have a cache mistake during entry locked, after 1 cycle of cache mistake judgment is lost, an external memory is accessed.

12.4. Register

12.4.1 Instruction Cache Capacity Setting Register

This register is to set how much KB of the 4 KB is used as an instruction cache.

ISIZE (8 bit)	7	6	5	4	3	2	1	0	Initial value
Addr. 0000 03C7 _H	-	-	-	-	-	-	SIZE1	SIZE0	---- --10 _B
	-	-	-	-	-	-	R/W	R/W	

- ISIZE [bits 1 to 0] : **SIZE1** and **SIZE0**

These bits set the cache capacity. The combination of the settings determines the I-Cache/I-RAM size, and the address map in RAM mode as shown in section 12.5. [Address Map \(Page No.278\)](#). When the cache size is changed, be sure to flush the cache and release the entry lock before turning on the cache.

Table 12.4-1 I-Cache Size Setting

CACHE Size Register		
SIZE1	SIZE0	Size
0	0	1 KB
0	1	2 KB
1	0	4 KB (initial value)
1	1	Setting disabled

12.4.2 Instruction Cache Control Register

ICHCR (I-CacHe Control Register) controls a behavior of an instruction cache.

The writing to ICHCR does not affect a cache behavior of an instruction fetched within subsequent 3 cycles..

ICHCR (8 bit)	7	6	5	4	3	2	1	0	Initial value
Addr.0000 03E7 _H	RAM	-	GBLK	ALFL	EOLK	ELKR	FLSH	ENAB	0-00 0000 _B
	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	

- [Bit 7] **RAM**: RAM Mode

Setting this bit to "1" causes the cache to operate in RAM mode.

By placing the cache in RAM mode, the cache RAM is mapped on an address map while the cache is enabled with the ENAB bit set to "1". For details, see section 12.5. [Address Map \(Page No.278\)](#).

- [Bit 6] Reserved bit

The writing is ignored. The read value is undefined.

- [Bit 5] **GBLK**: Global lock bit

All entries are locked.

When GBLK="1", a valid entry in the cache is not updated on a mistake. However, an invalid subblock is updated. At the time, instruction/data fetch behavior is performed in the same way as on unlocked.

- [Bit 4] **ALFL**: Auto lock fail bit

When an entry already locked is to be locked, ALFL is set to "1".

When an entry update during auto-locked is to be performed for an entry already locked, despite the user

intention, a new entry is not locked during caching. This bit is referred for a program debug. It is cleared with "0" writing.

- [Bit 3] **EOLK**: Entry auto lock bit
For each entry in an instruction cache, this bit switches valid/invalid of the auto-lock.
For the entry accessed (on a mistake only) with EOLK= "1", the entry lock bit in the cache tag is set to 1 with the hardware, and then the entry is locked. After that, the locked entry is out of target for updating on a cache mistake. However, an invalid subblock is updated. For secure lock, flush the cache once, and then set this bit.
- [Bit 2] **ELKR**: Entry lock release bit
Specify clearing for entry lock bits in all cache tags.
In the next cycle where ELKR=1 is set, entry lock bits in all cache tags are cleared to 0. However, the description of this bit is retained only for 1 clock cycle, and this bit is cleared to 0 for 2 clock cycle and later.
- [Bit 1] **FLSH**: Flush bit
A flush for an instruction cache is specified.
When FLSH="1", the content of the cache is flushed. However, the description of this bit is retained only for 1 clock cycle, and this bit is cleared to "0" for 2 clock cycle and later.
- [Bit 0] **ENAB**: Enable bit
Enable/Disable of the instruction cache is switched.
When ENAB="0", the instruction cache is in disabled status, and the instruction access from CPU is executed directly for external bus without the cache.
In the disabled status, the content of the cache is saved.

12.5. Address Map

12.5.1 Allocation within 4 KB RAM for Cache

MB91460 series has 4 KB RAM for the instruction cache.
The setting allows a selection from 4/2/1 KB for the area to be used as an instruction cache.

Since the cache has 2 ways, the allocation per way is 2 KB.
Each way is allocated with 2/1/0.5 KB depending on the cache area setting.
The remaining RAM is used as I-RAM.
If the instruction cache is prohibited, all areas are used as I-RAM.

The 2K RAM of one way are allocated as shown in [Figure 12.5-1](#).
When the cache area is restricted, I-RAM is allocated to the upper side.

Figure 12.5-1 Memory Allocation for Cache Capacities

Address	Instruction cache enabled: Set to 4KB	Instruction cache enabled: Set to 2KB	Instruction cache enabled: Set to 1KB	Instruction cache disabled
000 _H	CACHE RAM (2KB)	CACHE RAM (1KB)	CACHE RAM (512B)	I-RAM (2KB)
1FF _H				
200 _H		I-RAM (1KB)	I-RAM (1.5KB)	
3FF _H				
400 _H				
7FF _H				

12.5.2 Area on Memory Map

The allocation addresses of the instruction cache are 010000_H to 01FFFF_H.

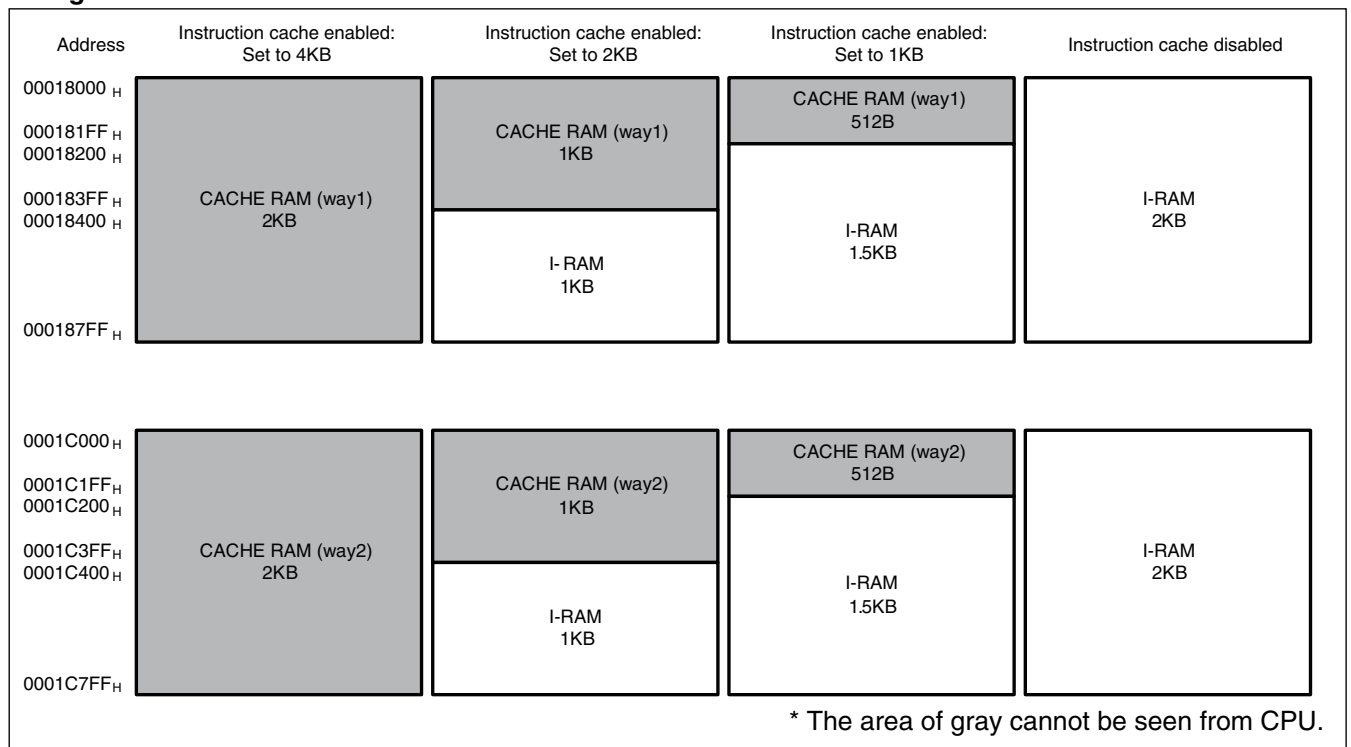
The allocation is described below specifically:

- 010000_H to 013FFF_H : TAG RAM (way1)
- 014000_H to 017FFF_H : TAG RAM (way2)
- 018000_H to 01BFFF_H : CACHE RAM(way1) / I-RAM
- 01C000_H to 01FFFF_H : CACHE RAM(way2) / I-RAM

How TAG-RAM and 4 K RAM for caching are allocated on the memory map is shown below.

The areas allocated to TAG RAM and instruction cache do not exist on the memory map. The address map for the time is shown in [Figure 12.5-2](#).

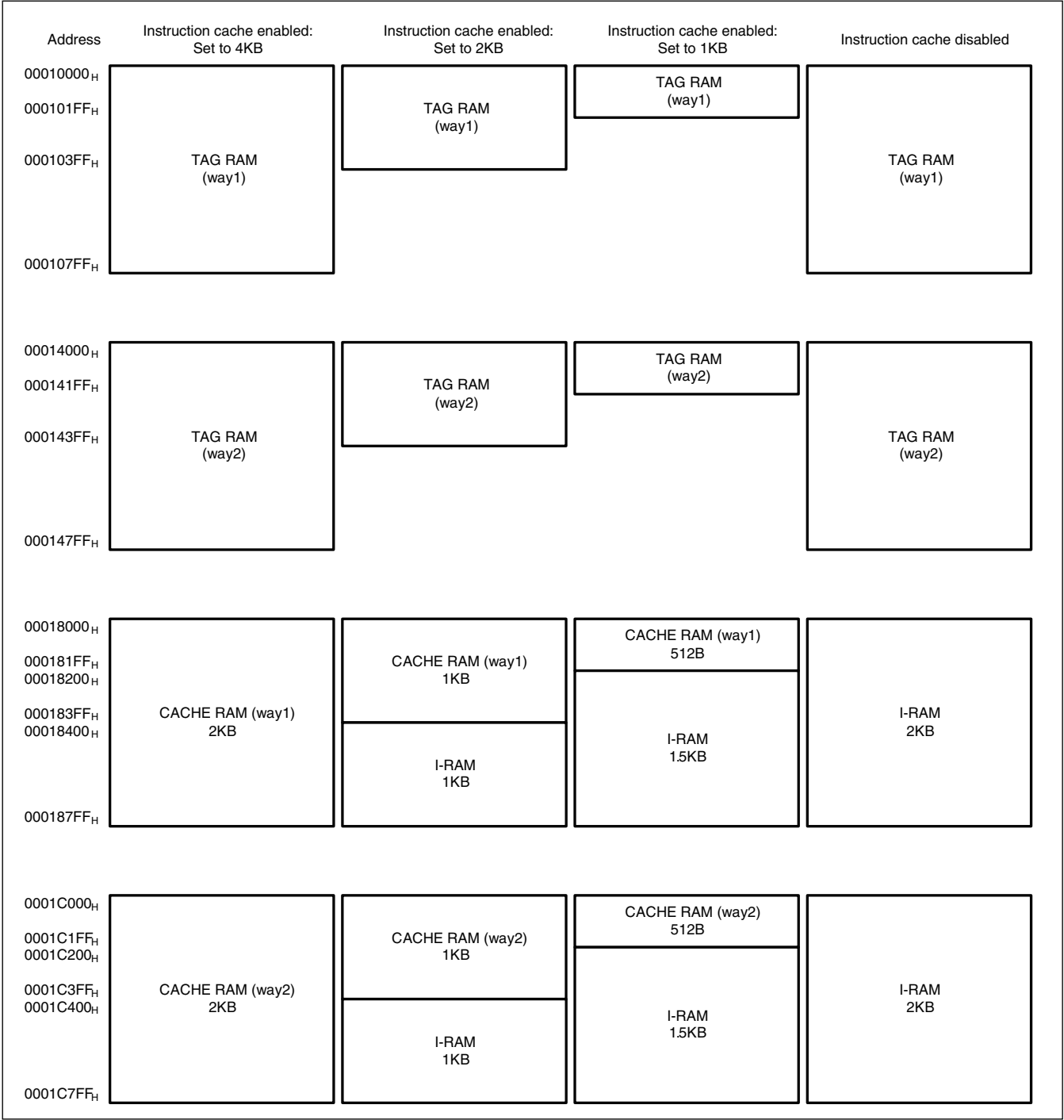
Figure 12.5-2 I-RAM allocation when ICHCR.RAM=0



The figure above shows the status where I-RAM only can be seen from the CPU.

If RAM mode bit (ICHCR:RAM) is set to "1", TAG-RAM and the area allocated to the instruction cache also appear. The address map at the time is shown in [Figure 12.5-3](#).

Figure 12.5-3 RAM allocation when ICHCR.RAM=1



The access judgment within each area is executed with lower bit.
Therefore, there is an image where all areas are filled with the RAM mirror area.
Each mirror image is shown in [Figure 12.5-4](#) and [Figure 12.5-5](#).

- TAG RAM

The TAG-RAM mirror allocation is shown in [Figure 12.5-4](#).

The mirror areas are allocated close together depending on cache areas.

The map is for ICHCR:RAM = "1". For ICHCR:RAM = "0", everything disappears.

Note that the access is enabled for any mirror area.

The way2 has the same way.

Figure 12.5-4 TAG-RAM mirror when ICHCR.RAM=1

Address	Instruction cache enabled: Set to 4KB	Instruction cache enabled: Set to 2KB	Instruction cache enabled: Set to 1KB	Instruction cache disabled
00010000 _H	TAG RAM (way1)	TAG RAM (way1)	TAG RAM (way1)	TAG RAM (way1)
000101FF _H			TAG RAM mirror (way1)	
00010200 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
000103FF _H			TAG RAM mirror (way1)	
00010400 _H			TAG RAM mirror (way1)	
000105FF _H			TAG RAM mirror (way1)	
00010600 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
000107FF _H			TAG RAM mirror (way1)	
00010800 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
000109FF _H			TAG RAM mirror (way1)	
00010A00 _H			TAG RAM mirror (way1)	
00010BFF _H			TAG RAM mirror (way1)	
00010C00 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
00010DFF _H			TAG RAM mirror (way1)	
00010E00 _H			TAG RAM mirror (way1)	
00010FFF _H			TAG RAM mirror (way1)	
00011000 _H		TAG RAM (way1)	TAG RAM (way1)	
000111FF _H			TAG RAM mirror (way1)	
00011200 _H	TAG RAM (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM (way1)
000113FF _H			TAG RAM mirror (way1)	
00011400 _H			TAG RAM mirror (way1)	
000115FF _H			TAG RAM mirror (way1)	
00011600 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
000117FF _H			TAG RAM mirror (way1)	
00011800 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
000119FF _H			TAG RAM mirror (way1)	
00011A00 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
00011BFF _H			TAG RAM mirror (way1)	
00011C00 _H			TAG RAM mirror (way1)	
00011DFF _H			TAG RAM mirror (way1)	
00011E00 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
00011FFF _H			TAG RAM mirror (way1)	
00012000 _H			TAG RAM mirror (way1)	
000121FF _H			TAG RAM mirror (way1)	
00012200 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
000123FF _H			TAG RAM mirror (way1)	
00012400 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
000125FF _H			TAG RAM mirror (way1)	
00012600 _H			TAG RAM mirror (way1)	
000127FF _H			TAG RAM mirror (way1)	
00012800 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
000129FF _H			TAG RAM mirror (way1)	
00012A00 _H	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)	TAG RAM mirror (way1)
00012BFF _H			TAG RAM mirror (way1)	
00012C00 _H			TAG RAM mirror (way1)	
00012DFF _H			TAG RAM mirror (way1)	
00012E00 _H		TAG RAM mirror (way1)	TAG RAM mirror (way1)	
00012FFF _H			TAG RAM mirror (way1)	

- CACHE RAM / I-RAM

The Cache-RAM / I-RAM mirror allocation is shown in [Figure 12.5-5](#).

A mirror appears for each 2 KB uniformly regardless of cache areas.

The map is for ICHCR:RAM = "1". For ICHCR:RAM = "0", all of CACHE RAM disappears.

Note that the access is enabled for any mirror area.

The way2 has the same way.

Figure 12.5-5 Cache RAM / I-RAM mirror

Address	Instruction cache enabled: Set to 4KB	Instruction cache enabled: Set to 2KB	Instruction cache enabled: Set to 1KB	Instruction cache disabled
00018000 _H 000181FF _H 00018200 _H	CACHE RAM (way1) 2KB	CACHE RAM (way1) 1KB	CACHE RAM (way1) 512B	I-RAM 2KB
000183FF _H 00018400 _H		I-RAM 1KB	I-RAM 1.5KB	
000187FF _H 00018800 _H	CACHE RAM (way1) mirror 2KB	CACHE RAM (way1) mirror 1KB	CACHE RAM (way1) mirror 512B	I-RAM mirror 2KB
000189FF _H 00018A00 _H		I-RAM mirror 1KB	I-RAM mirror 1.5KB	
00018BFF _H 00018C00 _H	CACHE RAM (way1) mirror 2KB	CACHE RAM (way1) mirror 1KB	CACHE RAM (way1) mirror 512B	I-RAM mirror 2KB
00018FFF _H 00019000 _H		I-RAM mirror 1KB	I-RAM mirror 1.5KB	
000191FF _H 00019200 _H	CACHE RAM (way1) mirror 2KB	CACHE RAM (way1) mirror 1KB	CACHE RAM (way1) mirror 512B	I-RAM mirror 2KB
000193FF _H 00019400 _H		I-RAM mirror 1KB	I-RAM mirror 1.5KB	
000197FF _H 00019800 _H	CACHE RAM (way1) mirror 2KB	CACHE RAM (way1) mirror 1KB	CACHE RAM (way1) mirror 512B	I-RAM mirror 2KB
0001B5FF _H 0001B600 _H		I-RAM mirror 1KB	I-RAM mirror 1.5KB	
0001B7FF _H 0001B800 _H	CACHE RAM (way1) mirror 2KB	CACHE RAM (way1) mirror 1KB	CACHE RAM (way1) mirror 512B	I-RAM mirror 2KB
0001BFFF _H		I-RAM mirror 1KB	I-RAM mirror 1.5KB	

12.5.3 TAG-RAM and Cache RAM Configuration

The tag of each block and cache data are stored in the same off-set location as the corresponding cache address.

For example, for the cache target area of Block0, lower 11 bits are within 000_H to 00F_H.

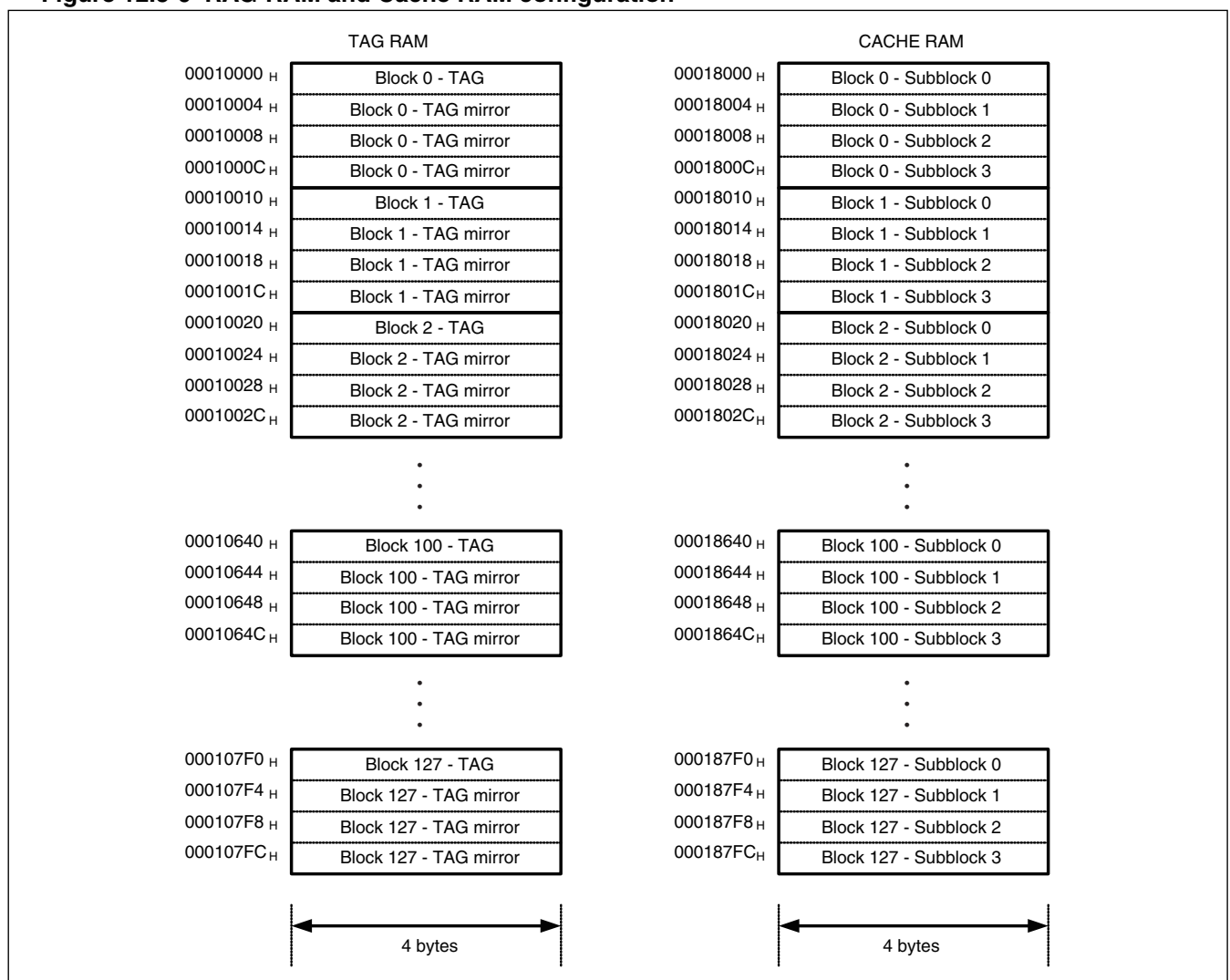
In this case, the tag of Block0 is stored to Number 000_H of TAG-RAM, and the cache data is stored to 000_H, 004_H, 008_H, and 00C_H of the cache RAM.

In the same way, for the cache target area of Block100, lower 11 bits are within 640_H to 64F_H.

In this case, the tag of Block100 is stored to the number 640_H of TAG-RAM, and the cache data is stored to 640_H, 644_H, 648_H, and 64C_H of the cache RAM.

The data map for way1 TAG-RAM and cache RAM is shown in [Figure 12.5-6](#).

Figure 12.5-6 TAG-RAM and Cache RAM configuration



For way2, only addresses differ, and the sequence is same.

12.6. Operating Modes

12.6.1 Cache Status in Various Operating Modes

Disable/flush indicates the status when the bit only is changed with a bit manipulation instruction, etc.

Table 12.6-1 I-Cache Status in various operating modes

		Shortly after Reset	Disable (ENAB = 0)	Flush
Cache memory		Content undefined	Last status retained Rewriting impossible during disabling	Last status retained
Tag	Address tag	Content undefined	Last status retained Rewriting impossible during disabling	Last status retained
	Subblock valid bit	Content undefined	Last status retained Rewriting impossible during disabling	Last status retained
	LRU	Content undefined	Last status retained Rewriting impossible during disabling	Last status retained
	Entry lock bit	Content undefined	Last status retained Rewriting impossible during disabling	Last status retained (Entry lock release is needed.)
	Tag valid bit	Content undefined	Last status retained Flush possible during disabling	All entries invalid
RAM		Normal mode	Last status retained Flush possible during disabling	Last status retained
Control register	Global lock	Unlock	Last status retained Rewriting possible during disabling	Last status retained
	Auto lock fail	No fail	Last status retained Rewriting possible during disabling	Last status retained
	Entry auto lock	Unlock	Last status retained Rewriting possible during disabling	Last status retained
	Entry lock release	No release	Last status retained Rewriting possible during disabling	Last status retained
	Enable	Disable	Disable	Last status retained
	Flush	No flush	Last status retained Rewriting possible during disabling	Flush in the immediate cycle after memory access. After that, the bit returns to 0.

12.6.2 Cache Entry Update

Cache entries are updated as shown in the following table.

Table 12.6-2 I-Cache Entry Update

	Unlock	Lock
Hit	Not updated	Not updated.
Mistake	Updated	For a tag mistake, no update. For a subblock disabled, update.

12.6.3 Cacheable areas in the instruction cache

- The instruction cache can cache data only in external bus space.

Figure 12.6-1 I-Cacheable Area

Address	F-bus disabled	Address	F-bus enabled
00000000	Direct area	00000000	Direct area
00010000	IRAM	00010000	IRAM
00020000	D-bus RAM	00020000	D-bus RAM
00030000		00030000	
00040000		00040000	F-bus
00100000	Cache area	00100000	Cache area
FFFFFFFF		FFFFFFFF	

- Even when the contents of external memory are updated by DMA transfer, the instruction cache does not refresh its contents to be coherent with the new contents of the memory. In this case, flush the cache to give it coherence.
- Each chip select area can be set as a non-cacheable area. The penalty for this is one cycle compared to the cache off state.

12.7. Setting Method When I-cache is Used

(1) Initialization

Before I-Cache is brought into use, the cache content must be cleared.

Set the FLUSH bit and ELKR bit of the register to "1" to erase the past data.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000110,r1  // FLSH bit (1 bit)
                        // ELKR bit (2 bits)
stb      r1,@r0          // Writing to the register
```

Now the cache is initialized.

(2) The cache is enabled (ON).

Set ENAB bit to "1" to enable I-Cache.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000001,r1  // ENAB bit (0 bit)
stb      r1,@r0          // Writing to the register
```

The following instruction accesses are taken into the cache.

The cache validating can be executed at the same time with the cache initialization.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000111,r1  // ENAB bit (0 bit)
                        // FLSH bit (1 bit)
                        // ELKR bit (2 bits)
stb      r1,@r0          // Writing to the register
```

(3) The cache is disabled (OFF).

Set ENAB bit to "0" to disable I-Cache.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000000,r1  // ENAB bit (0 bit)
stb      r1,@r0          // Writing to the register
```

With this status (same with the status after reset), the cache is practically nought and dose nothing.

For the processing where the cache overhead is concerned, the cache OFF might improve the processing performance.

(4) All contents of the cache are locked.

Lock the instruction which is in I-Cache currently not to be updated.

Set the GBLK bit of the register to "1".

If the ENAB bit also is not set to "1", the cache is turned to OFF, and the locked instruction within the cache is not used.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00100001,r1  // ENAB bit (0 bit)
                        // GBLK bit (5 bits)
stb      r1,@r0          // Writing to the register
```

(5) Specified instructions are locked to the cache.

To lock specified instructions group (such as a subroutine) to the cache, set the EOLK bit to "1" before those instructions are executed.

The locked instruction is accessed as a high-speed internal ROM.The lock is enabled from the instruction next to the

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00001001,r1  // ENAB bit (0 bit)
                        // EOLK bit (3 bits)
stb      r1,@r0          // Writing to the register
```

stb instruction or later depending on the waiting number of memory.

Set the EOLK bit to "0" when instructions group you wish to lock is completed.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000001,r1  // ENAB bit (0 bit)
                        // EOLK bit (3 bits)
stb      r1,@r0          // Writing to the register
```

(6) The cache lock is released.

Release the lock information of the instruction locked in (5) above.

```
ldi      #_ICHCR,r0      // I-Cache control register address
ldi      #0b00000000,r   // ENAB bit (0 bit)
stb      r1,@r0          // Writing to the register
ldi      #0b00000100,r   // ELKR bit (2 bits)
stb      r1,@r0          // Writing to the register
```

Since the lock information only is released, the locked instructions are replaced with new instructions in series depending on the LRU bit status.

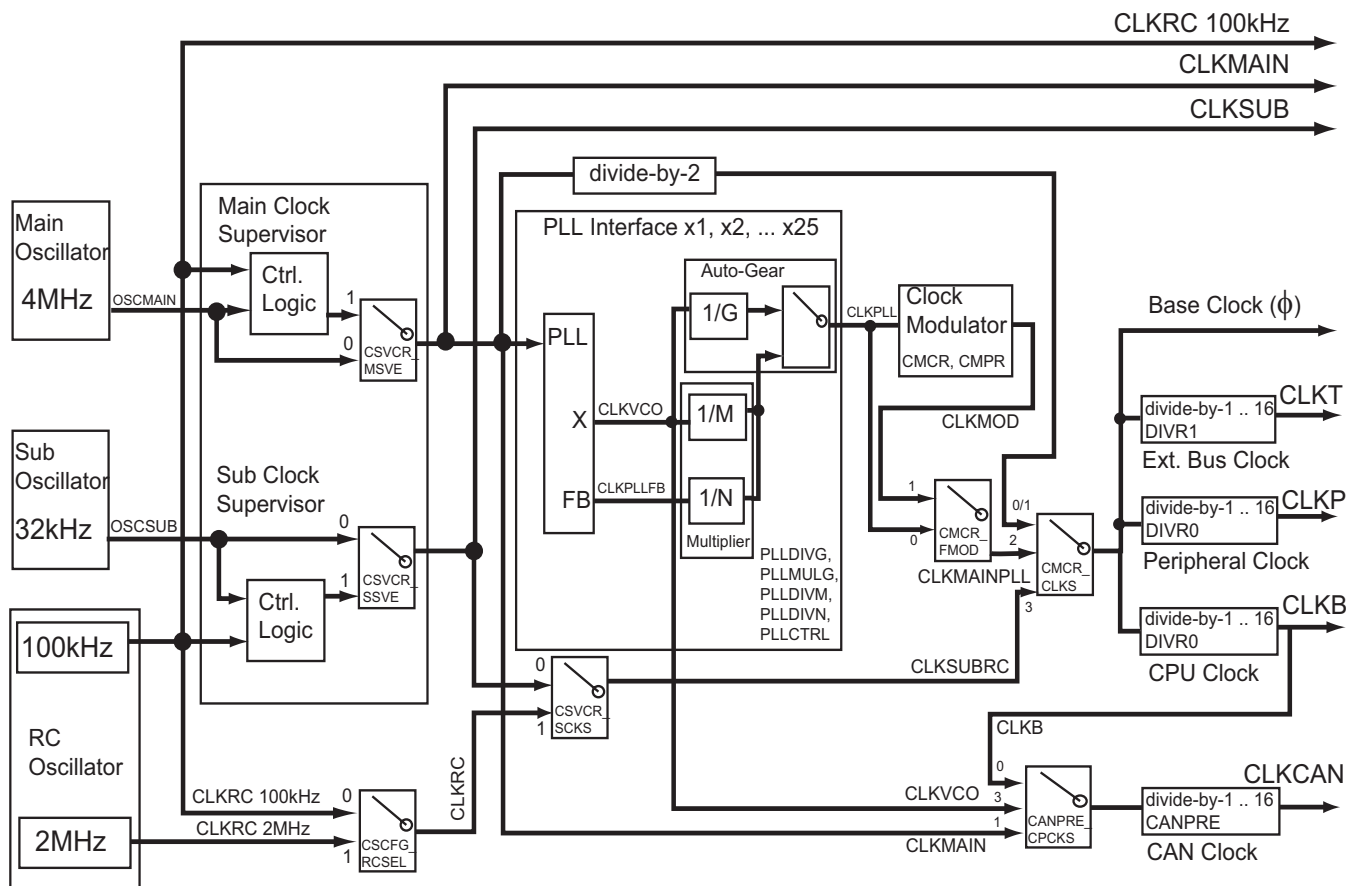
MB91460 Series

Chapter 13 Clock Control

13.1. Overview

The clock control circuit consists of the source oscillator, base clock generator, and operating clock generator.

Figure 13.1-1 Block Diagram of Clock Distribution



Explanation to PLL Interface: 1/G means $1/(PLLDIVG + 1)$

1/M means $1/(PLLDIVM + 1)$

1/N means $1/(PLLDIVN + 1)$

13.2. Features

■ Source oscillation

- Main clock (CLKMAIN): 4MHz (typical value)
Input from the X0/X1 pins and used as the high speed clock
- Sub clock (CLKSUB): 32.768kHz (typical value)
Input from the X0A/X1A pins and used as the low speed clock
- RC clock (CLKRC 100kHz): 100kHz (typical value)
RC oscillator and used as the low speed clock

■ Base clock (ϕ): Selectable from 3 different clocks

- Main PLL (programmable): $\text{CLKMAIN} \times (\text{MxN})/\text{M}$ (either modulated or unmodulated)
- CLKMAIN divided by 2
- CLKSUBRC: CLKSUB or CLKRC

■ Operating clocks: Selectable from 16 different speeds

- **CPU clock (CLKB): $\phi/1, /2, /3, /4, /5, /6, /7, /8, \dots, /16$**

The clock used by the CPU, internal memory, and internal buses. The circuits that use this clock are as follows.

- CPU, internal RAM, internal ROM, bit search module,
- I bus, D bus, F bus, X bus
- On-chip debug support unit (DSU)

- **Peripheral clock (CLKP): $\phi/1, /2, /3, /4, /5, /6, /7, /8, \dots, /16$**

This clock is used by the peripheral functions and peripheral bus. The circuits that use this clock are as follows.

- Peripheral bus
- Clock controller (bus interface unit only)
- Interrupt controller
- I/O ports
- External interrupt inputs, UART, 16-bit timer, and similar peripheral functions

- **External bus clock (CLKT): $\phi/1, /2, /3, /4, /5, /6, /7, /8, \dots, /16$**

This clock is used by the external bus expansion interface. The circuits that use this clock are as follows.

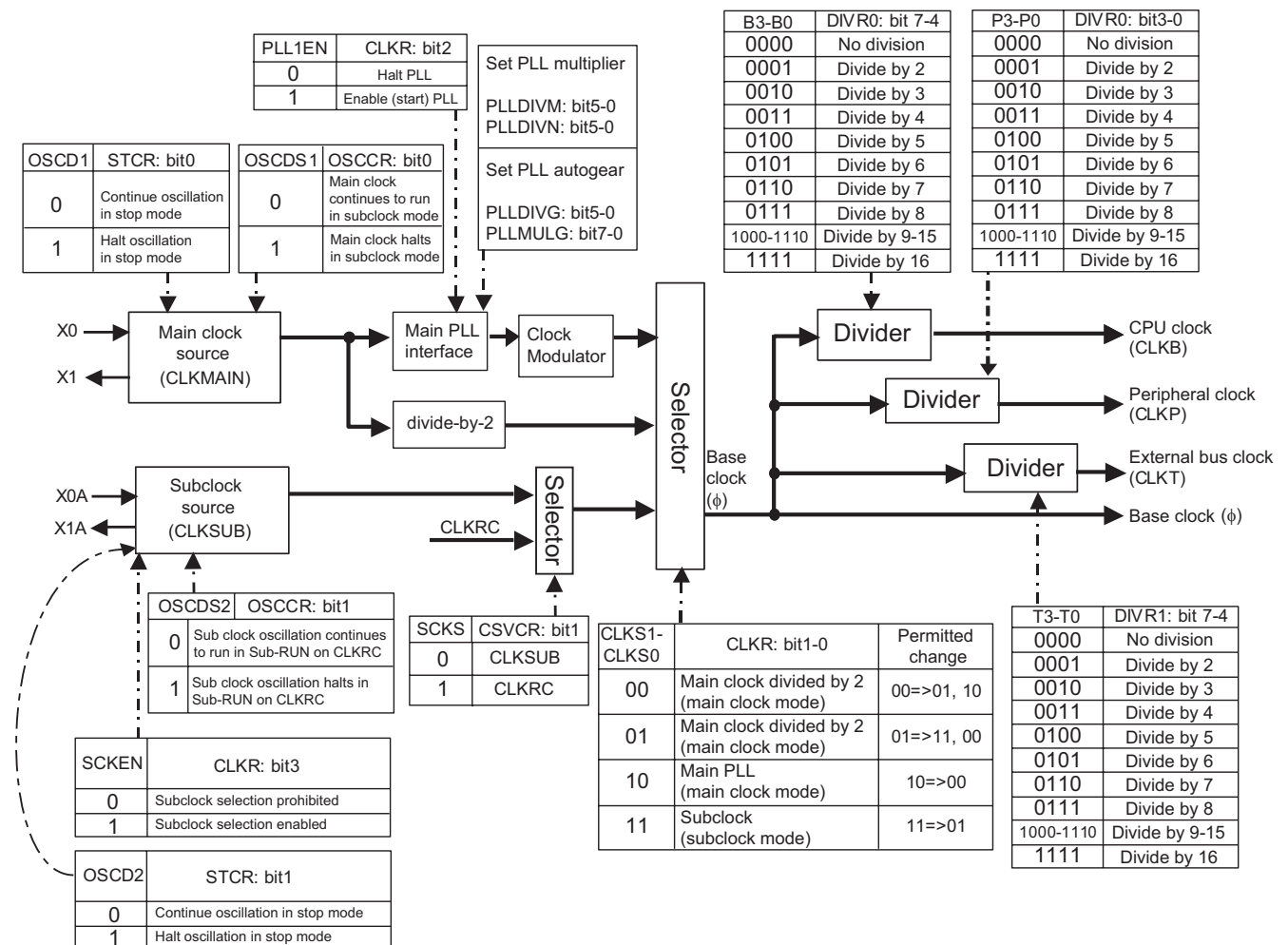
- External bus expansion interface
- External CLK output

MB91460 Series

13.3. Configuration

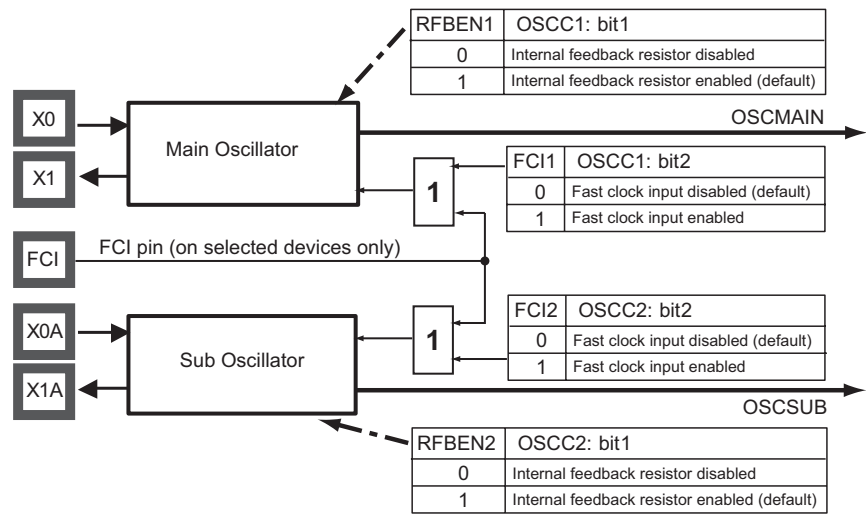
13.3.1 Clock Configuration

Figure 13.3-1 Configuration Diagram of the clock controller



13.3.2 Oscillator Configuration

Figure 13.3-2 Configuration Diagram of the Main- and Sub-Oscillators



13.4. Registers

Figure 13.4-1 Clock Control Register List

Clock Control:

Address	Bit	7	6	5	4	3	2	1	0		
00481H		STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	STCR	Standby control
00484H		---	---	---	---	SCKEN	PLL1EN	CLKS1	CLKS0	CLKR	Clock source control
00486H		B3	B2	B1	B0	P3	P2	P1	P0	DIVR0	Operating clock division setting 0
00487H		T3	T2	T1	T0	---	---	---	---	DIVR1	Operating clock division setting 1
004AEH		EDSUEN	PLLLOCK	RCSEL	MONCLKI	CSC3	CSC2	CSC1	CSC0	CSCFG	Clock source configuration
004CCH		---	---	---	---	---	---	OSCD2	OSCD1	OSCCR	Oscillation control

Oscillator Parameter Control:

Address	Bit	7	6	5	4	3	2	1	0		
00494H		---	---	---	---	---	FCI1	RFBEN1	reserved	OSCC1	Main oscillator control
00496H		---	---	---	---	---	FCI2	RFBEN2	reserved	OSCC2	Sub oscillator control

MB91460 Series**13.4.1 STCR: Standby Control Register**

This register controls the standby modes and the oscillator behaviour in STOP mode.

- **STCR: Address 0481h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	
0	0	1	1	0	0	1	1	Initial value (INITX pin input)
0	0	1	1	X	X	1	1	Initial value (Watchdog reset)
0	0	X	1	X	X	X	X	Initial value after RST (Software reset)
R/W	R/W	R/W	R1, W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit7-2: STOP, SLEEP, HIZ, SRST, OS1, OS0
 - About these bits, please refer to [Chapter 10 Standby \(Page No.253\)](#).

- Bit1: Sub oscillator oscillation halt

OSCD2	Operation of Sub oscillator during STOP state
0	Continue oscillation
1	Halt oscillation

- Bit0: Main oscillator oscillation halt

OSCD1	Operation of Main oscillator during STOP state
0	Continue oscillation
1	Halt oscillation

13.4.2 CLKR: Clock Source Control Register

Selects the clock source for the base clock used to run the MCU and controls the PLL.

• **CLKR: Address 0484h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	-	-	SCKEN	PLL1EN	CLKS1	CLKS0	
X	X	X	X	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-4: Reserved bit.
 - Always write “0” to this bit. The read value is the value written.
- Bit3: Sub clock select enable

SCKEN	Function
0	Stop Sub clock oscillator. [Initial value]
1	Enable Sub clock selection

- Modifying the Sub clock selection enable bit (SCKEN) while the Sub clock is selected as the clock source (CLKS[1:0] = “11”) is prohibited (result is not guaranteed). Only modify the setting when the Main clock is selected.
(See the explanation for the clock source selection bits (CLKS[1:0]) for details of how to change the clock source.)
- Bit2: Enable Main PLL operation

PLL1EN	Function
0	Halt Main PLL (Initial value)
1	Enable Main PLL operation

- Modifying the Main PLL operation enable bit (PLL1EN) while the Main PLL is selected as the clock source (CLKS[1:0] = “10”) is prohibited.
- Modifying the Main PLL operation enable bit (PLL1EN) while the clock autogear function is active (gear up or gear down) is prohibited. Always check the gear status flags before changing the PLL state (see chapter “[Clock Auto-Gear Up/Down](#)” on P. 319).
- If the Main clock oscillation is halted (STCR.OSCD1 = “1”), the Main PLL halts during STOP state even if the PLL enable bit (PLL1EN) is set to “1”. If Main PLL operation is enabled (PLL1EN = “1”), the Main clock operates using the PLL after recovering from STOP state.
(See the explanation for the clock source selection bits (bits 1-0:) for details of changing the clock source.)
- Bit1-0: Clock source selection

CLKS1	CLKS0	Clock source setting	Mode
0	0	Main clock input from X0/X1 divided by 2 (Initial value)	Main clock mode
0	1	Main clock input from X0/X1 divided by 2	Main clock mode
1	0	Main PLL	Main clock mode
1	1	Sub clock	Sub clock mode

- When changing the clock mode, the value of CLKS0 must not be modified while CLKS1 is “1”.

MB91460 Series

The table below lists the cases in which the CLKS1- CLKS0 bits may be modified.

- After the clock mode bit CLKS1 is set from “1” to “0” by write command the CLKS1 is changed not immediately to “0”.
- There is no setting to select the Sub clock divided by 2.
- After setting “11_B” (Sub clock), insert one or more NOP instructions.
- Selecting the Sub clock as the clock source is prohibited while the Sub clock selection enable bit (SCKEN) is “0”. (See table for details.)

Table 13.4-1 Allowed transitions from CLKS1 and CLKS0

Modify permitted	Modify prohibited
“00” → “01” or “10”	“00” → “11”
“01” → “11” or “00”	“01” → “10”
“10” → “00”	“10” → “01” or “11”
“11” → “01”	“11” → “00” or “10”

Example: To select the Sub clock after an INIT reset, first write “01_B” and then write “11_B” (Sub clock).

(See “13.8. Caution (Page No.311)”.)

13.4.3 DIVR0: Clock Division Setting Register 0

Sets the division ratio for the clocks used for internal device operation.

- **DIVR0: Address 0486h (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
B3	B2	B1	B0	P3	P2	P1	P0	
0	0	0	0	0	0	1	1	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Sets up the clock for the CPU and internal buses (CLKB), and the clock for the peripheral circuits and peripheral bus (CLKP).
- Bit7-4: CLKB division selection

B3-B0	CPU clock (CLKB) division ratio
0000	$\Phi/1$ (initial value)
0001	$\Phi/2$
0010	$\Phi/3$
0011	$\Phi/4$
0100	$\Phi/5$
0101	$\Phi/6$
0110	$\Phi/7$
0111	$\Phi/8$
1000	$\Phi/9$
1001	$\Phi/10$
1010	$\Phi/11$
1011	$\Phi/12$
1100	$\Phi/13$
1101	$\Phi/14$
1110	$\Phi/15$
1111	$\Phi/16$

- **Do not change the division ratio with B3-B0 if current CLKB frequency is equal or above 80MHz!**
- Sets the clock division ratio for the clock used by the CPU, internal memory, and internal buses (CLKB). The 16 options listed in the table are available.
- Do not set a division ratio that exceeds the maximum operating frequency of the device.
- Bit3-0: CLKP division selection

P3-P0	Peripheral clock (CLKP) division ratio
0000	$\Phi/1$
0001	$\Phi/2$
0010	$\Phi/3$
0011	$\Phi/4$ (initial value)
0100	$\Phi/5$
0101	$\Phi/6$
0110	$\Phi/7$

MB91460 Series

0111	$\Phi/8$
1000	$\Phi/9$
1001	$\Phi/10$
1010	$\Phi/11$
1011	$\Phi/12$
1100	$\Phi/13$
1101	$\Phi/14$
1110	$\Phi/15$
1111	$\Phi/16$

- Sets the clock division ratio for the clock used by the peripheral circuits and peripheral bus (CLKP). The 16 options listed in the table are available.
- Do not set a division ratio that exceeds the maximum operating frequency of the MCU.

13.4.4 DIVR1: Clock Division Setting Register 1

Sets the division ratio for the clocks used for internal device operation.

- **DIVR1: Address 0487h (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
T3	T2	T1	T0	—	—	—	—	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

Sets the clock division ratio (relative to the base clock) for the clock used by the external bus interface (CLKT).

- Bit7-4: CLKT division selection

T3-T0	External bus clock (CLKT) division ratio
0000	$\Phi/1$ (initial value)
0001	$\Phi/2$
0010	$\Phi/3$
0011	$\Phi/4$
0100	$\Phi/5$
0101	$\Phi/6$
0110	$\Phi/7$
0111	$\Phi/8$
1000	$\Phi/9$
1001	$\Phi/10$
1010	$\Phi/11$
1011	$\Phi/12$
1100	$\Phi/13$
1101	$\Phi/14$
1110	$\Phi/15$
1111	$\Phi/16$

- Sets the clock division ratio for the clock used by the external bus interface (CLKT). The 16 options listed in the table are available.
- Do not set a division ratio that exceeds the maximum operating frequency of the external bus interface.
- If the CLKP division selection bits are modified, the new division ratio applies from the next clock after the setting is modified.
- Bit3-0: Reserved bits
 - Always write “0” to these bits. The read value is the value written.

MB91460 Series**13.4.5 CSCFG: Clock Source Configuration Register**

- **CSCFG: Address 04AEh (Access: Byte)**

7	6	5	4	3	2	1	0	bit
EDSUEN	PLLLOCK	RCSEL	MONCKI	CSC3	CSC2	CSC1	CSC0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit7: EDSU/MPU Enable

EDSUEN	Function
0	The clock for the EDSU/MPU module is disabled [Initial value]
1	The clock for the EDSU/MPU module is enabled

- bit6: PLL Lock

PLLLOCK	Function
0	PLL is in the un-locked state
1	PLL is in the locked state

This bit shows that the Main PLL is enabled (CLKR:PLL1EN='1') and is locked.

This bit may switch to '0' and back to '1' unexpectedly during normal PLL operation. It is **not** recommended to poll this bit at PLL startup. Please use the Timebase Timer interrupt to wait for PLL lock time at startup.

- bit5: CLKRC Selector

RCSEL	Function
0	CLKRC is set to 100kHz [Initial value]
1	CLKRC is set to 2MHz

- The selected oscillation frequency is supplied to Clock Control Unit (for Sub clock Operation) and Flash Security Unit (change the oscillation to 2MHz for faster CRC generation). Hardware Watchdog (RC based Watchdog), Real Time Clock, Calibration Unit, LCD and Clock Supervisor Module are always supplied with 100kHz independent of this setting.

- bit4: Clock Monitor MONCLK inverter

MONCKI	Function
0	MONCLK mark level is low [Initial value]
1	MONCLK mark level is high

See chapter “[Chapter 48 Clock Monitor \(Page No.1123\)](#)” about information about this function.

- Bit3: Clock Source Selection for LCD Controller

CSC3	Function
0	LCD Controller is sourced by Sub Oscillator
1	LCD Controller is sourced by RC Oscillator (100kHz)

- Bit 2: Clock Source Selection for Sub clock calibration

CSC2	Function
0	Sub clock Calibration is sourced by Sub Oscillator
1	Sub clock Calibration is sourced by RC Oscillator(100kHz)

-

- Bit1-0: Clock Source Selection for RTC

CSC1-CSC0	Function
00	Real Time Clock is sourced by Main Oscillator
01	Real Time Clock is sourced by Sub Oscillator
10	Real Time Clock is sourced by RC Oscillator(100kHz)
11	Setting prohibited

MB91460 Series

13.4.6 OSCCR: Oscillation Control Register

This register controls the Main clock oscillation and the Sub clock oscillation in Sub clock mode.

- **OSCCR: Address 04CCh (Access: Byte)**

7	6	5	4	3	2	1	0	bit
–	–	–	–	–	–	OSCD2	OSCD1	
X	X	X	X	X	X	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit7-2: Undefined bit

Writing does not affect the operation. The read value is undefined.

- bit1: Stop Sub clock oscillation in Sub-RUN on CLKRC.

OSCD2	Operation when written to	Read value meaning
0	Does not halt Sub clock oscillation during Sub-RUN on CLKRC.	Sub clock mode can be selected after the oscillation stabilization time elapses.
1	Halt Sub clock oscillation during Sub-RUN on CLKRC.	Selecting Sub clock mode is prohibited.

- When the Sub clock oscillation is selected (CSVCR.SCKS='0'), specifying the Sub clock oscillation to halt during Sub-RUN state (OSCD2="1") is prohibited.

(See “[13.8. Caution \(Page No.311\)](#)”.)

- bit0: Stop Main clock oscillation in Sub clock mode.

OSCD1	Operation when written to	Read value meaning
0	Does not halt Main clock oscillation during Sub clock mode.	Main clock mode can be selected after the oscillation stabilization time elapses.
1	Halt Main clock oscillation during Sub clock mode.	Selecting Main clock mode is prohibited.

- When the Main clock oscillation is selected (CLKS[1:0]="00", "01", "10"), specifying the Main clock to halt during Sub clock mode (OSCD1="1") is prohibited.

(See “[13.8. Caution \(Page No.311\)](#)”.)

- The following table lists the settings which are necessary to operate with the different clock sources. Secondly the table shows, when the clock sources can be switched off.

	CLKS[1:0]	SCKS	switch off Main oscillation	switch off Sub oscillation
Main oscillator	00, 01, 10	-	prohibited	prohibited
Sub oscillator	11	0	OSCD1 = 1	prohibited
RC oscillator	11	1	OSCD1 = 1	OSCD2 = 1

13.4.7 OSCC1: Main Oscillator Control Register

This register controls the parameters of the Main Oscillator.

- **OSCC1: Address 0494h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
–	–	–	–	–	FCI1	RFBEN1	reserved	
X	X	X	X	X	0	1	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit7-3: Undefined bits

Writing does not affect the operation. The read value is undefined.

- bit2: Fast Clock Input mode for Main Oscillator

FCI1	Function
0	Main oscillator is in oscillation mode (default)
1	Main oscillator is in Fast Clock Input mode

- This bit is initialized to "0" by each initialisation reset (INIT)
- This bit selects the oscillation or Fast Clock Input mode of the Main Oscillator.
- This bit must be set to "0" when connecting an oscillator (crystal/resonator) to the Main Oscillator pins X0 and X1. It is also possible to connect an external clock with low frequency to the oscillator X0/X1 pins in this mode.
- For inputting an external clock with higher frequency, this bit must be set to "1" before the device is switched to external (main) clock. The external clock can be connected to pin X0.
- The frequency ranges for both settings are described in the datasheet.
- This feature is not available on all devices. See datasheet for more details.

- bit1: Internal Feedback Resistor for Main Oscillator

RFBEN1	Function
0	The internal feedback resistor is disabled
1	The internal feedback resistor is enabled (default)

- This bit is initialized to "1" by each initialisation reset (INIT)
- This bit enables the internal feedback resistor in the Main Oscillator.

- bit0: reserved bit

- always write "0" to this bit

MB91460 Series

13.4.8 OSCC2: Sub Oscillator Control Register

This register controls the parameters of the Sub Oscillator.

- **OSCC2: Address 0496h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
–	–	–	–	–	FCI2	RFBEN2	reserved	
X	X	X	X	X	0	1	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit7-3: Undefined bits

Writing does not affect the operation. The read value is undefined.

- bit2: Fast Clock Input mode for Sub Oscillator

FCI2	Function
0	Sub oscillator is in oscillation mode (default)
1	Sub oscillator is in Fast Clock Input mode

- This bit is initialized to "0" by each initialisation reset (INIT)
- This bit selects the oscillation or Fast Clock Input mode of the Sub Oscillator.
- This bit must be set to "0" when connecting an oscillator (crystal/resonator) to the Sub Oscillator pins X0A and X1A. It is also possible to connect an external clock with low frequency to the oscillator X0A/X1A pins in this mode.
- For inputting an external clock with higher frequency, this bit must be set to "1" before the device is switched to external (main) clock. The external clock can be connected to pin X0A.
- The frequency ranges for both settings are described in the datasheet.
- This feature is not available on all devices. See datasheet for more details.

- bit1: Internal Feedback Resistor for Sub Oscillator

RFBEN2	Function
0	The internal feedback resistor is disabled
1	The internal feedback resistor is enabled (default)

- This bit is initialized to "1" by each initialisation reset (INIT)
- This bit enables the internal feedback resistor in the Sub Oscillator.

- bit0: reserved bit

- always write "0" to this bit

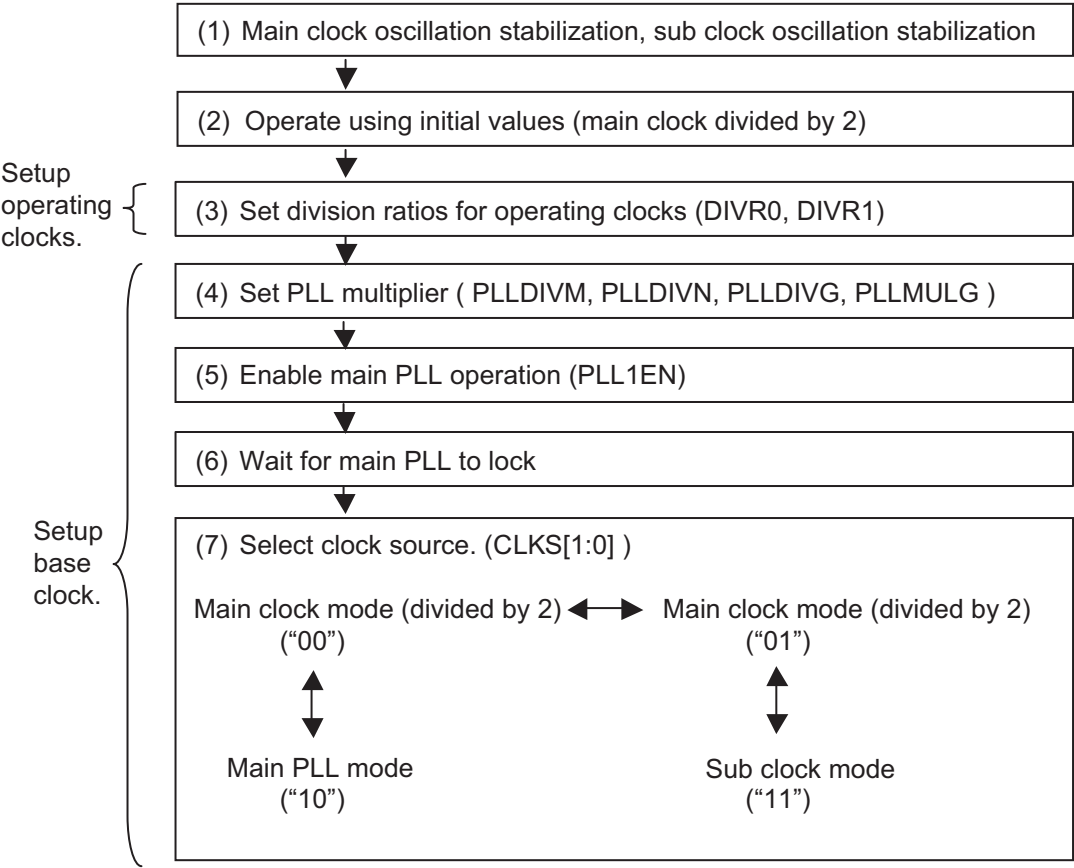
13.5. Operation

This section describes how to setup and switch between clocks.

13.5.1 Clock Setup Sequence

Figure 13.5-1 shows an example how to set up the clocks after setting initialization reset (INIT).

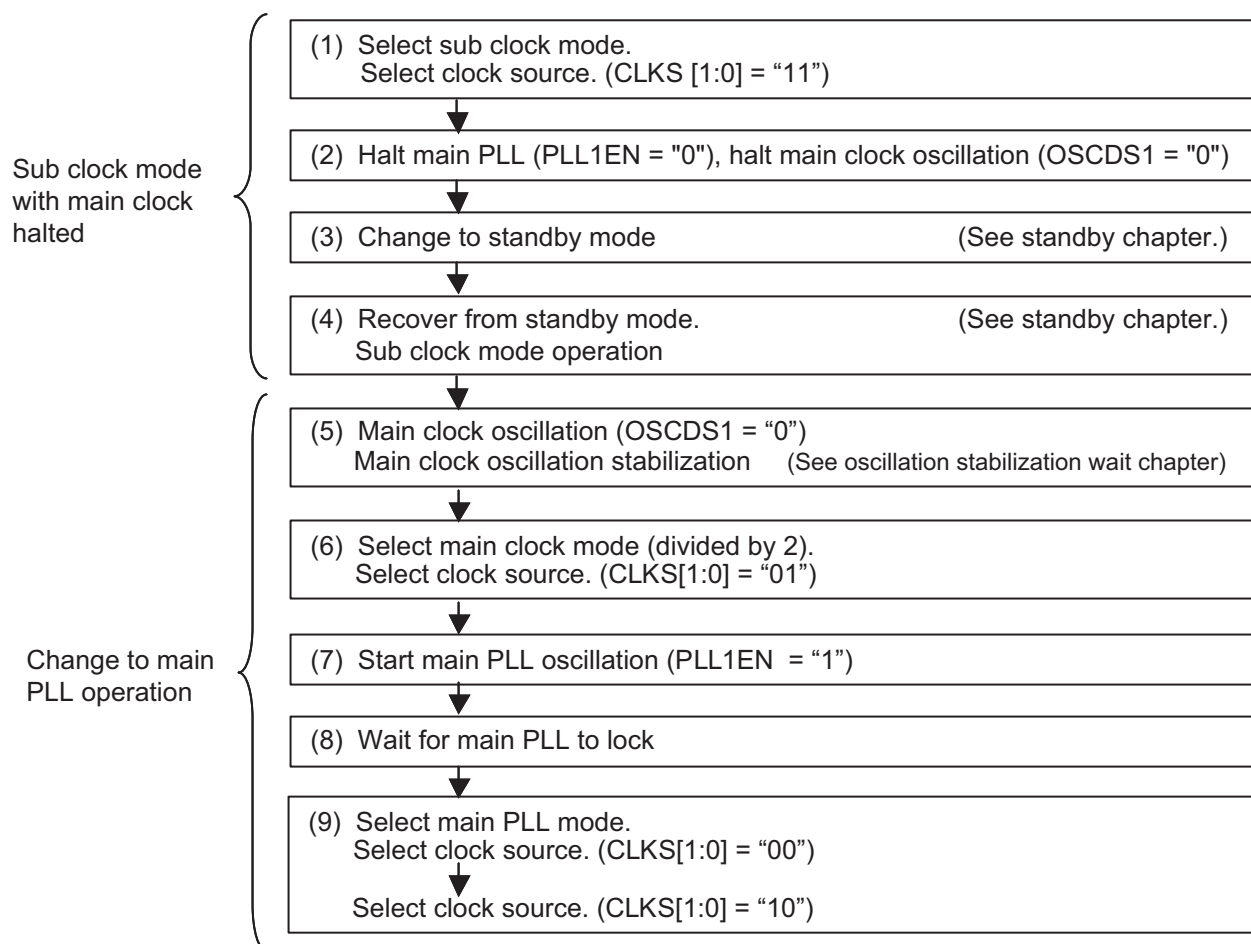
Figure 13.5-1 Clock Setup Sequence (Example)



13.5.2 Halting and Restarting the Main Clock Oscillation During Sub Clock Mode

Figure 13.5-2 shows an example how to switch from main to sub clock and halt the main clock during sub clock (and standby) mode, as well as the switching back from sub clock to main PLL.

Figure 13.5-2 Halting and Restarting the Main Clock Oscillation During Sub Clock Mode (Example)



13.5.3 Notes

■ Main PLL control

After settings initialization reset (INIT) initialization, the Main PLL oscillation is stopped. While it is stopped, the output of the Main PLL cannot be selected as the clock source.

When the program operation starts, set the multiplier of the PLL to be used as the clock source, enable it, and switch the source clock after the PLL lock wait time elapses. For the PLL lock wait time, use of a time base timer interrupt is recommended.

The Main PLL cannot be stopped while the output of the Main PLL is selected as the clock source (writing to the register has no effect). To stop the Main PLL upon transition to STOP state, reselect the Main clock divided by two as source clock before stopping the PLL.

If STCR.OSCD1 or STCR.OSCD2 of the standby control register is set to "1" to stop oscillation in STOP state, the Main PLL automatically stops when the device enters STOP state. There is no need to stop the Main PLL (CLKR.PLL1EN="0") explicitly beforehand.

The Main PLL also restarts automatically when recovering from STOP state. When the oscillation is not set to stop during STOP state (STCR.OSCD1="0"), the Main PLL does not stop automatically. In this case, stop the Main PLL explicitly (CLKR.PLL1EN="0") before changing to STOP state.

■ Main PLL multiplier

To change the Main PLL multiplier setting, change it before or as soon as the Main PLL is enabled after program execution starts. After changing the multiplier setting, wait for the Main PLL lock time before switching the clock source. For the Main PLL lock wait time, it is recommended to use a time base timer interrupt.

To change the Main PLL multiplier setting during operation, first change the clock source to a clock other than the Main PLL. As described above, after changing the multiplier setting, wait for the Main PLL lock time before changing the clock source.

■ Clock division

The clocks used to drive the internal operation of the device allow division ratios relative to the base clock to be set independently for each clock. This function allows the optimum operating frequency to be selected for each circuit.

The division ratios are set in the operating clock division setting registers (DIVR0 and DIVR1). These registers contain 4-bit settings that specify the ratio for each clock. The division ratio relative to the base clock equals (register value+1). The duty ratio is always 50, even if an odd numbered division ratio is set.

If the setting is modified, the new setting becomes valid at the next rising edge of the clock.

The division ratio setting is not initialized if an operation initialization reset (RST) occurs, but the settings from before the reset are retained. The ratio setting is only initialized by a settings initialization reset (INIT). When changing the clock source from its initial setting to a high speed clock, always set the division ratio first.

An upper-limit frequency for the operation is set for each clock.

Device operation is not guaranteed if the combination of clock source selection, Main PLL multiplier setting, and division ratio setting results in a frequency higher than the maximum permitted frequency.

13.6. Settings

Table 13.6-1 Settings for Operating at 1/2 of the Main Clock

Setting	Setting register	Setting procedure
Clock source selection	Clock source control register (CLKR)	See 13.7.3

Table 13.6-2 Settings for Operating Using the Main PLL

Setting	Setting register	Setting procedure
Main PLL operation enable	Clock source control register (CLKR)	See 13.7.1
Clock source selection		See 13.7.3

Table 13.6-3 Settings for Operating Using the Sub clock

Setting	Setting register	Setting procedure
Sub clock selection enable	Clock source control register (CLKR)	See 13.7.1
Clock source selection		See 13.7.3

Table 13.6-4 Settings for Selecting the Division Ratio for the Operating Clocks

Setting	Setting register	Setting procedure
Clock source selection	Clock source control register (CLKR)	See 13.7.3
Operating clock division ratio selection	Operating clock division setting registers (DIVR0, DIVR1)	See 13.7.4

13.7. Q & A

13.7.1 How to enable or disable the clock operation

- There is no operation enable bit for the Main clock.
(Halting the oscillation in Sub clock mode or STOP state is handled separately.)
- Main PLL operation is enabled by the Main PLL operation enable bit (CLKR.PLL1EN).

Operation	Main PLL operation enable bit (PLL1EN)
To halt the Main PLL	Set to "0".
To enable operation of the Main PLL	Set to "1".

Initially, the PLL is halted and therefore PLL operation must be enabled and the PLL started after setting the PLL multiplier ratio.

- The Sub clock on the MB91460 does not halt and therefore no corresponding operation enable bit is provided.
Instead, the Sub clock selection enable bit (CLKR.SCKEN) is used.

Operation	Sub clock selection enable bit (SCKEN)
Sub clock selection prohibited	Set to "0".
To enable selection of the Sub clock	Set to "1".

13.7.2 How to select the Main PLL multiplier ratio

- The PLL multiplier can be set by using the PLL interface registers PLLDIVM and PLLDIVN (see [Chapter 14 Main PLL Interface \(Page No.313\)](#)).

13.7.3 How to select the operating clock source

Use the clock source selection bits (CLKR.CLKS[1:0]) to select Main clock divided by 2, Main PLL, or the Sub clock as the operating clock source.

Operating clock source	Clock source selection bits (CLKS[1:0])
To change from the initial value to the Main clock divided by 2	Set initial values "00" or "01".
To change from the initial value to the Main PLL	Change from the initial values "00" to "10".
To change from the initial value to the Sub clock	First change from the initial values "00" to "01", and then to "11".
To change from the Sub clock to the Main clock divided by 2	Change from "11" to "01".
To change from the Sub clock to Main PLL	First change from "11" to "01", next set "00", and then set "10".
To change from Main PLL to the Main clock divided by 2	Change from "10" to "00".
To change from Main PLL to the Sub clock	First change from "10" to "00". Wait until CLKS[1]='0'. Next set "01", and then set "11".

MB91460 Series

13.7.4 How to set the operation clock division ratios

- CPU clock setting

The CPU clock setting is set using the CLKB division ratio selection bits (DIVR0.B[7:4]).

Base clock (ϕ) division ratio	CLKB division ratio selection bits(B[7:4])	Example frequency	
		When $\phi = 32\text{MHz}$	When $\phi = 16\text{MHz}$
no division	Set to "0000".	CLKB = 32.0MHz	CLKB = 16.0MHz
2	Set to "0001".	CLKB = 16.0MHz	CLKB = 8.00MHz
3	Set to "0010".	CLKB = 10.6MHz	CLKB = 5.33MHz
4	Set to "0011".	CLKB = 8.00MHz	CLKB = 4.00MHz
5	Set to "0100".	CLKB = 6.40MHz	CLKB = 3.20MHz
6	Set to "0101".	CLKB = 5.33MHz	CLKB = 2.66MHz
7	Set to "0110".	CLKB = 4.57MHz	CLKB = 2.28MHz
8	Set to "0111".	CLKB = 4.00MHz	CLKB = 2.00MHz
16	Set to "1111".	CLKB = 2.00MHz	CLKB = 1.00MHz

- Peripheral clock setting

The peripheral clock setting is set using the CLKP division ratio selection bits (DIVR0.P[3:0]).

Base clock (ϕ) division ratio	CLKP division ratio selection bits (P[3:0])	Example frequency	
		When $\phi = 32\text{MHz}$	When $\phi = 16\text{MHz}$
no division	Set to "0000".	CLKP = 32.0MHz	CLKP = 16.0MHz
2	Set to "0001".	CLKP = 16.0MHz	CLKP = 8.00MHz
3	Set to "0010".	CLKP = 10.6MHz	CLKP = 5.33MHz
4	Set to "0011".	CLKP = 8.00MHz	CLKP = 4.00MHz
5	Set to "0100".	CLKP = 6.40MHz	CLKP = 3.20MHz
6	Set to "0101".	CLKP = 5.33MHz	CLKP = 2.66MHz
7	Set to "0110".	CLKP = 4.57MHz	CLKP = 2.28MHz
8	Set to "0111".	CLKP = 4.00MHz	CLKP = 2.00MHz
16	Set to "1111".	CLKP = 2.00MHz	CLKP = 1.00MHz

- Setting for the external bus clock

The setting for the external bus clock is set using the CLKT division ratio selection bits (DIVR1.T[3:0]).

Base clock (ϕ) division ratio	CLKT division ratio selection bits (T[7:4])	Example frequency	
		When $\phi = 32\text{MHz}$	When $\phi = 16\text{MHz}$
no division	Set to "0000".	CLKT = 32.0MHz	CLKT = 16.0MHz
2	Set to "0001".	CLKT = 16.0MHz	CLKT = 8.00MHz
3	Set to "0010".	CLKT = 10.6MHz	CLKT = 5.33MHz
4	Set to "0011".	CLKT = 8.00MHz	CLKT = 4.00MHz
5	Set to "0100".	CLKT = 6.40MHz	CLKT = 3.20MHz
6	Set to "0101".	CLKT = 5.33MHz	CLKT = 2.66MHz
7	Set to "0110".	CLKT = 4.57MHz	CLKT = 2.28MHz
8	Set to "0111".	CLKT = 4.00MHz	CLKT = 2.00MHz
16	Set to "1111".	CLKT = 2.00MHz	CLKT = 1.00MHz

13.7.5 How to halt the Main clock in Sub clock mode

Set using the “halt Main clock oscillation in Sub clock mode” bit (OSCCR.OSCDS1).

Operation in Sub clock mode	“Halt Main clock oscillation in Sub clock mode” bit (OSCDS1)
To not halt the Main clock	Set to “0”.
To halt the Main clock	Set to “1”.

(See “13.8. Caution (Page No.311)”.)

13.7.6 How to halt the Sub clock in Sub-RUN on CLKRC

Set using the “halt Sub clock oscillation in Sub-RUN on CLKRC” bit (OSCCR.OSCDS2).

Operation in Sub-RUN state	“Halt Sub clock oscillation in Sub-RUN on CLKRC” bit (OSCDS2)
To not halt the Sub clock	Set to “0”.
To halt the Sub clock	Set to “1”.

(See “13.8. Caution (Page No.311)”.)

13.7.7 How to halt the Sub clock in Main clock mode

Operation in Main clock mode	“Halt Sub clock oscillation in Main clock mode” bit (OSCD2)
To not halt the Sub clock	Set to “0”.
To halt the Sub clock	Set to “1”.

13.8. Caution

- Operation is not guaranteed if the clock source selection, Main PLL multiplier setting, and division ratio setting result in a frequency that exceeds the maximum.
- It is necessary to follow the sequence in which the clock source selection is set or modified.
- When the Main clock oscillation is set to halt during Sub clock mode (OSCDS1 = "1"), selecting the Main clock (CLKS[1:0])="00", "01", or "10") is prohibited. To select the Main clock, set (OSCDS1="0") and then change to the Main clock after waiting for the Main clock oscillation to stabilize. Use the Main clock oscillation stabilization wait timer to provide the wait time. See "[Chapter 22 Main Oscillation Stabilization Timer \(Page No.405\)](#)" for details.
- When the Sub clock oscillation is set to halt during Sub clock mode (OSCDS2 = "1"), selecting the Sub clock (CSVCR.SCKS="0") is prohibited. To select the Sub clock, set (OSCDS2="0") and then change to the Sub clock after waiting for the Sub clock oscillation to stabilize. Use the Sub clock oscillation stabilization wait timer to provide the wait time. See "[Chapter 23 Sub Oscillation Stabilization Timer \(Page No.417\)](#)" for details.
- When the Main clock oscillation is halted (OSCDS1 = "1") or the Sub clock oscillation is halted (OSCDS2 = "1") an oscillation stabilization wait time (for Main clock or Sub clock) is also required if a reset (INIT) occurs that switches the clock source to the Main clock. In this case, operation after the reset is not guaranteed if the wait time set in the oscillation stabilization time selection bits (STCR.OS[1:0]) does not satisfy the oscillation stabilization time requirement for the Main clock.
Always set the oscillation stabilization time selection bits (STCR.OS[1:0]) to a value that provides an adequate oscillation stabilization time for the Main clock.
In the case of an INIT reset triggered by the INITX pin, the "L" level input must be maintained for long enough for the Main clock oscillation to stabilize.
See "[Chapter 18 Timebase Counter \(Page No.363\)](#)" and "[Chapter 22 Main Oscillation Stabilization Timer \(Page No.405\)](#)" for details of the oscillation stabilization wait.
- When changing to STOP state, the Main PLL must either be halted or de-selected. Either set the Main clock oscillation halt bit (STCR.OSCD1 = "1") to halt automatically or change the operating clock to Main clock divided by two before changing to STOP state.

Chapter 14 PLL Interfaces (Main PLL, E-Ray, USB)

14.1. Overview

On MB91460 series devices, there may be up to 4 PLLs:

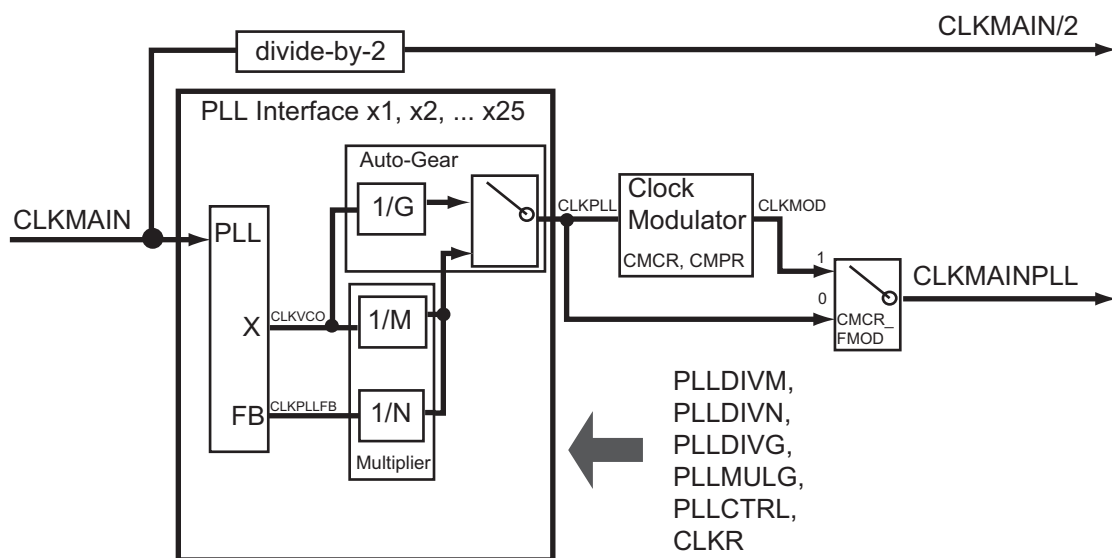
- The Main PLL can supply the CPU and resource clocks as well as the CAN clocks and is implemented on all devices.
- If a FlexRay (E-Ray) module is implemented, there is a separate E-Ray PLL.
- If an USB module is implemented, there is a separate USB PLL.
- If MediaLB / I²S are implemented, there is a separate MediaLB PLL.

The interfaces of MAIN, E-Ray and USB PLL are described in this chapter. For the MediaLB PLL, please refer to [Chapter 59 MediaLB Clock Generation/Bus Interface \(Page No.1407\)](#).

14.2. Main PLL Interface

The simplified block diagram in [Figure 14.2-1](#) shows the integration of the main PLL and the Main PLL Interface with the multiplier control logic (PLLDIVM (1/M), PLLDIVN (1/N) for basic frequency multiplication and PLLDIVG (1/G) for clock auto gear).

Figure 14.2-1 Block diagram of Main PLL interface



14.2.1 Main PLL Features

- Free programmable divide-by-M counter (1/M) in the range of 1..16
- Free programmable divide-by-N counter (1/N) in the range of 1..64
- Clock auto gear up/down function to prevent voltage drops and surges

14.2.2 Main PLL Frequency calculation

- CLKB frequency is determined by :

$$\text{CLKB} = [\text{CLKMAIN} * (\text{PLLDIVM_DVM}+1) * (\text{PLLDIVN_DVN}+1)] / [(\text{PLLDIVM_DVM}+1) * (\text{DIVR0_B}+1)]$$

- CLKP frequency is determined by :

$$\text{CLKP} = [\text{CLKMAIN} * (\text{PLLDIVM_DVM}+1) * (\text{PLLDIVN_DVN}+1)] / [(\text{PLLDIVM_DVM}+1) * (\text{DIVR0_P}+1)]$$

- CLKT frequency is determined by :

$$\text{CLKT} = [\text{CLKMAIN} * (\text{PLLDIVM_DVM}+1) * (\text{PLLDIVN_DVN}+1)] / [(\text{PLLDIVM_DVM}+1) * (\text{DIVR1_T}+1)]$$

14.2.3 Clock Auto-Gear Up/Down

To avoid voltage drops and surges when switching the clock source from oscillator to high frequency PLL/DLL output (or vice versa), a clock smooth gear-up and gear-down circuitry is implemented with the PLL interface.

The main functionality is implemented using two divide-by counters (divide-by-M and divide-by-G counter), where one supplies the PLL feedback always with the target frequency (divide-by-M counter), and the other (divide-by-G counter) which increases the frequency from a programmable frequency division given by the divide-by-G setting (DIVG) up to the target frequency given by the divide-by-M setting (DIVM), or decreases the frequency from the divide-by-M setting (DIVM) down to the programmable end frequency (DIVG).

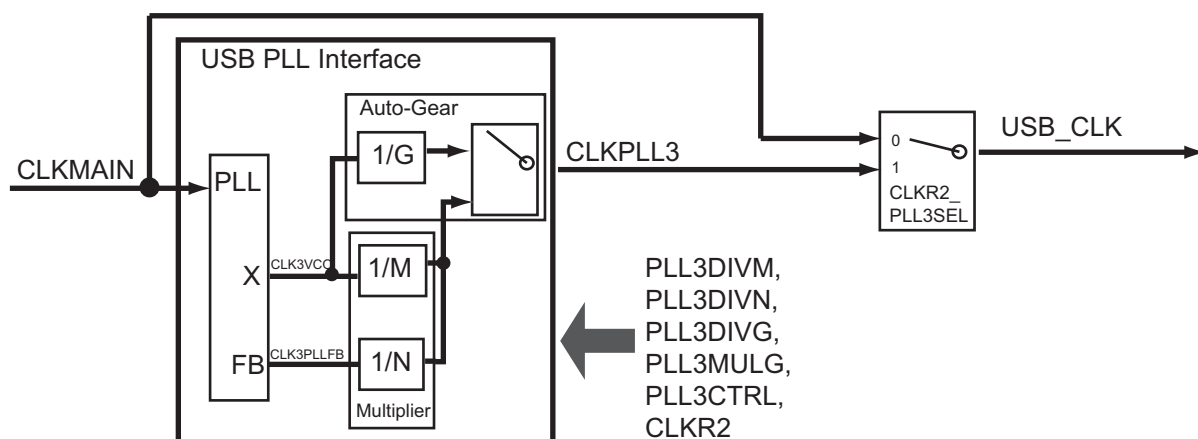
In this sense only a setting of $\text{DIVG} > \text{DIVM}$ is a valid clock gear specification to scale the system clock from slower frequencies to faster frequencies (when gearing up) and from faster frequencies to slower ones (when gearing down).

The frequency steps are performed in multiple of the PLL output frequency, e.g. the setting of: Oscillator = 4MHz, $M = 2$, $N = 20$ (which is a frequency multiplication of $M * N = 40$ with PLL output = 160MHz and frequency output to C-Unit = 80MHz).

The gear divider can be set to any even divider, in this example it is $G = 20$, which causes the following gear-up when switching from oscillator to PLL:

1. step : 1 cycle of 8.0MHz (8.0MHz equals 20 cycles of the PLL output)
 2. step : 2 cycles of 8.4MHz (8.4MHz equals 19 cycles of the PLL output)
 3. step : 3 cycles of 8.8MHz (8.8MHz equals 18 cycles of the PLL output)
 - :
 17. step : 17 cycles of 40.0MHz (40.0MHz equals 4 cycles of the PLL output)
 18. step : 18 cycles of 53.3MHz (53.3MHz equals 3 cycles of the PLL output)
 19. step : 19 cycles of 80.0MHz (80.0MHz equals 2 cycles of the PLL output)
- > Target frequency reached by transition to last step (here from 18. to 19.)

Each step can be multiplied by setting a multiplication value in the gear multiplier register. The duration from generating the start frequency up to reaching the target frequency can be calculated by the following formula:

Figure 14.3-2 USB PLL Interface

14.3.2 E-Ray and USB PLL Interface Features

Both interfaces implemented the following features:

- Free programmable divide-by-M counter (1/M) in the range of 1..16
- Free programmable divide-by-N counter (1/N) in the range of 1..64
- Clock auto gear up/down function to prevent voltage drops and surges

The E-Ray clock (ERAY_SCLK) can be selected from CLKB, Main PLL or E-Ray PLL.

The USB clock (USB_CLK) can use main oscillation or the USB PLL.

MB91460 Series**14.4. Registers****14.4.1 Register Overview****Table 14.4-1 PLL Interface Register Overview**

Address	7	6	5	4	3	2	1	0	Register	
00484 _H	-	-	-	-	SCKEN	PLL1EN	CLKS1	CLKS0	Main PLL	CLKR
0048C _H	-	-	-	-	DVM3	DVM2	DVM1	DVM0		PLLDIVM
0048D _H	-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0		PLLDIVN
0048E _H	-	-	-	-	DVG3	DVG2	DVG1	DVG0		PLLDIVG
0048F _H	MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0		PLLMULG
00490 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP		PLLCTRL
004DC _H	-	-	-	-	DVM3	DVM2	DVM1	DVM0	E-Ray PLL	PLL2DIVM
004DD _H	-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0		PLL2DIVN
004DE _H	-	-	-	-	DVG3	DVG2	DVG1	DVG0		PLL2DIVG
004DF _H	MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0		PLL2MULG
004E0 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP		PLL2CTRL
004E2 _H	SEL23	PLL3EN	PLL3SEL	SELP12	SCDBL	PLL2EN	CLKS1	CLKS0		CLKR2
004E4 _H	-	-	-	-	DVM3	DVM2	DVM1	DVM0	USB PLL	PLL3DIVM
004E5 _H	-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0		PLL3DIVN
004E6 _H	-	-	-	-	DVG3	DVG2	DVG1	DVG0		PLL3DIVG
004E7 _H	MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0		PLL3MULG
004E8 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP		PLL3CTRL

14.4.2 Main PLL Control Registers

Controls the Main PLL multiplier ratio (divide-by-M and divide-by-N) and the automatic clock gear up/down function.

■ CLKR: Address 0484h (Access: Byte)

7	6	5	4	3	2	1	0	bit
-	-	-	-	SCKEN	PLL1EN	CLKS1	CLKS0	
X	X	X	X	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

For CLKR description, please refer to section [13.4.2 CLKR: Clock Source Control Register \(Page No.294\)](#).

■ PLLDIVM: Address 048Ch (Access: Byte, Halfword, Word)

7	6	5	4	3	2	1	0	bit
-	-	-	-	DVM3	DVM2	DVM1	DVM0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	0	0	X	X	X	X	Initial value (Software reset)
R0/W0	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-4: Reserved bits. Always write “0” to these bits.
- Bit3-0: PLL divide-by-M selection

DVM3-DVM0	PLL output (CLKPLL)
0000	CLKVCO : 1 (no division)
0001	CLKVCO : 2 (division by 2)
0010	CLKVCO : 3 (division by 3)
0011	CLKVCO : 4 (division by 4)
0100	CLKVCO : 5 (division by 5)
0101	CLKVCO : 6 (division by 6)
0110	CLKVCO : 7 (division by 7)
.....
1111	CLKVCO : 16 (division by 16)

Note: Even though it is possible to select no division ratio (:1) for the divide-by-M counter it is not recommended. The resulting output clock will have an odd clock duty ratio (direct PLL output). Always select at least a division ratio > 1 and an even division ratio (:2, :4, :6, etc.).

Note: Even though it is possible to select an odd division ratio (:3, :5, :7, etc.) for the divide-by-M counter it is not recommended. The resulting output clock will have an odd clock duty ratio. Always select an even division ratio (:2, :4, :6, etc.).

Note: The register value can not be changed once PLL is selected as clock source (CLKS[1:0]=”10”). It is strongly recommended to disable the PLL (CLKR.PLL1EN=0) while or after changing the PLLDIVM and PLLDIVN registers and to enable the PLL (CLKR.PLL1EN=1) afterwards.

■ PLLDIVN: Address 048Dh (Access: Byte, Halfword, Word)

7	6	5	4	3	2	1	0	bit
-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	X	X	X	X	X	X	Initial value (Software reset)
R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-6: Reserved bits. The read value is always “0”.
- Bit5-0: PLL divide-by-N selection

DVN5-DVN0	feedback to PLL (CLKPLLFB)
000000	CLKPLL : 1 (no division)
000001	CLKPLL : 2 (division by 2)
000010	CLKPLL : 3 (division by 3)
000011	CLKPLL : 4 (division by 4)
000100	CLKPLL : 5 (division by 5)
000101	CLKPLL : 6 (division by 6)
000110	CLKPLL : 7 (division by 7)
000111	CLKPLL : 8 (division by 8)
.....
111111	CLKPLL : 64 (division by 64)

Note: The register value can not be changed once PLL is selected as clock source (CLKS[1:0]=”10”).

Note: It is strongly recommended to disable the PLL (CLKR.PLL1EN=0) while or after changing the PLLDIVM and PLLDIVN registers and to enable the PLL (CLKR.PLL1EN=1) afterwards.

■ PLLDIVG: Address 048Eh (Access: Byte, Halfword, Word)

7	6	5	4	3	2	1	0	bit
-	-	-	-	DVG3	DVG2	DVG1	DVG0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	0	0	X	X	X	X	Initial value (Software reset)
R0/W0	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-4: Reserved bits. Always write “0” to these bits.
- Bit3-0: PLL auto gear start/end divide-by-G selection

DVG3-DVG0	PLL output start/end frequency (CLKPLL with Auto-gear function)
0000	Auto gear disabled (initial value)
0001	CLKVCO : 2 (division by 2)
0010	CLKVCO : 3 (division by 3)
0011	CLKVCO : 4 (division by 4)
0100	CLKVCO : 5 (division by 5)

0101	CLKVCO : 6 (division by 6)
0110	CLKVCO : 7 (division by 7)
0111	CLKVCO : 8 (division by 8)
.....
1111	CLKVCO : 16 (division by 16)

Note: See section [14.2.3 Clock Auto-Gear Up/Down \(Page No.314\)](#) for detailed information on how to use this function.

Note: Even though it is possible to select an odd division ratio (:3, :5, :7, etc.) for the divide-by-G counter it is not recommended. Always select an even division ratio (:2, :4, :6, etc.).

Note: The register value can not be changed once PLL is selected as clock source (CLKS[1:0]="10").

■ PLLMULG: Address 048Fh (Access: Byte, Halfword, Word)

7	6	5	4	3	2	1	0	bit
MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See "[Meaning of Bit Attribute Symbols \(Page No.15\)](#)" for details of the attributes.)

- Bit7-0: PLL auto gear divide-by-G step multiplier selection

MLG5-MLG0	Divide-by-G step multiplier
00000000	Divide-by-G step x 1 (multiply by 1)
00000001	Divide-by-G step x 2 (multiply by 2)
00000010	Divide-by-G step x 3 (multiply by 3)
00000011	Divide-by-G step x 4 (multiply by 4)
00000100	Divide-by-G step x 5 (multiply by 5)
00000101	Divide-by-G step x 6 (multiply by 6)
00000110	Divide-by-G step x 7 (multiply by 7)
00000111	Divide-by-G step x 8 (multiply by 8)
.....
11111111	Divide-by-G step x 256 (multiply by 256)

Note: See section [14.2.3 Clock Auto-Gear Up/Down \(Page No.314\)](#) for detailed information on how to use this function.

Note: The register value can not be changed once PLL is selected as clock source (CLKS[1:0]="10").

■ PLLCTRL: Address 0490h (Access: Byte, Halfword, Word)

7	6	5	4	3	2	1	0	bit
-	-	-	-	IEDN	GRDN	IEUP	GRUP	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	0	0	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	RM1/W	R/W	RM1/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-4: Reserved bit The read value is always “0”.
- Bit3: Interrupt Request Enable Gear DOWN.

IEDN	Function
0	Gear DOWN interrupt request disabled [Initial value]
1	Gear DOWN interrupt request enabled

- Bit2: Interrupt Flag Gear DOWN.

GRDN	Function
0	Gear DOWN interrupt not active [Initial value]
1	Gear DOWN interrupt active

- While switching from clock source PLL to clock source oscillator this flag is set when the divide-by-G counter reaches the programmed end value.
- This bit is read as “1” at a Read-Modify-Write instructions. Writing “1” has no effect.
- Bit1: Interrupt Request Enable Gear UP.

IEUP	Function
0	Gear UP interrupt request disabled [Initial value]
1	Gear UP interrupt request enabled

- Bit0: Interrupt Flag Gear UP.

GRUP	Function
0	Gear UP interrupt not active [Initial value]
1	Gear UP interrupt active

- While switching from clock source oscillator to clock source PLL this flag is set when the divide-by-G counter reaches the end value defined by the divide-by-M counter.
- This bit is read as “1” at a Read-Modify-Write instructions. Writing “1” has no effect.

14.4.3 CLKR2 Register

The CLKR2 register is used to control the E-Ray PLL as well as the USB PLL.

If neither USB nor E-Ray are implemented on the device, this register does not exist and read access returns 0xFF.

■ CLKR2: Address 04E2h (Access: Byte)

15	14	13	12	11	10	9	8	bit
SELP23	PLL3EN	PLL3SEL	SELP12	SCDBL	PLL2EN	CLKS1	CLKS0	
0	0	0	0	0	0	0	0	Initial value (INIT)
X	X	X	X	X	X	X	X	Initial value (RST)
R/W	R/W	R,W	R/W	R0/W	R/W	R,W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit 7,4: SELP23, SELP12 Clock Monitor signal select.

SELP23	SELP12	Function
X	0	(default) The clock monitor outputs the signals of the Main PLL
0	1	The clock monitor outputs the signals of the E-Ray PLL
1	1	The clock monitor outputs the signals of the USB PLL

- These bits are cleared by INIT (settings initialization reset).
- If USB is not implemented, bit SELP23 does not exist and reads '0'.

- Bit 6: PLL3EN Enable USB PLL.

PLL3EN	Function
0	(default) USB PLL is disabled
1	USB PLL is enabled

- This bit is cleared by INIT (settings initialization reset).
- If USB is not implemented, this bit does not exist and reads '0'.
- This bit can only be written if the USB PLL is not clock source (PLL3SEL = '0')

- Bit 5: PLL3SEL USB clock source selection

PLL3SEL	Function
0	(default) The Main oscillator CLKMAIN is USB clock source
1	The USB PLL is USB clock source

- This bit is cleared by INIT (settings initialization reset).
- If USB is not implemented, this bit does not exist and reads '0'.
- This bit can only be written if the PLL is running (PLL3EN = '1').
- If this bit is written '0', a request to gear down the PLL is generated.
The bit reads '1' until this PLL gear down is finished.

- Bit 3: SCDBL E-Ray serial clock disable.

SCDBL	Function
0	(default) E-Ray clock is enabled
1	E-Ray clock is disabled

MB91460 Series

- This bit is cleared by INIT (settings initialization reset).
- The read value of this bit is always '0'.
- Bit 2: PLL2EN E-Ray PLL enable.

PLL2EN	Function
0	(default) E-Ray PLL is disabled
1	E-Ray PLL is enabled

- This bit is cleared by INIT (settings initialization reset).
- This bit can only be written if the E-RAY PLL is not clock source (CLKS1,CLKS0 != "10")
- Bit 1-0: CLKS[1:0] E-Ray clock (SCLK) source selection.

CLKS1	CLKS0	Function
0	0	(default) SCLK is operated with CLKB (coreclock)
0	1	SCLK is operated with Main PLL (baseclock)
1	0	SCLK is operated with FlexRay PLL (recommended setting)
1	1	Testmode only, do not set!

- Both bits are cleared by INIT (settings initialization reset).
- CLKS0 can be read and written by CPU.
- CLKS1 can only be written if PLL2EN is '1'.
- If CLKS1 is written '0', a request to gear down the PLL is generated.
The bit CLKS1 reads '1' until this PLL gear down is finished.

14.4.4 E-Ray PLL Registers

The following E-Ray PLL control registers

Address	7	6	5	4	3	2	1	0	Register	
004DC _H	-	-	-	-	DVM3	DVM2	DVM1	DVM0	E-Ray PLL	PLL2DIVM
004DD _H	-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0		PLL2DIVN
004DE _H	-	-	-	-	DVG3	DVG2	DVG1	DVG0		PLL2DIVG
004DF _H	MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0		PLL2MULG
004E0 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP		PLL2CTRL

have the same functionality like the appropriate registers of the Main PLL. Please refer to section [14.4.2 Main PLL Control Registers \(Page No.318\)](#).

14.4.5 USB PLL Registers

The following USB PLL Control registers

Address	7	6	5	4	3	2	1	0	Register	
004E4 _H	-	-	-	-	DVM3	DVM2	DVM1	DVM0	USB PLL	PLL3DIVM
004E5 _H	-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0		PLL3DIVN
004E6 _H	-	-	-	-	DVG3	DVG2	DVG1	DVG0		PLL3DIVG
004E7 _H	MLG7	MLG6	MLG5	MLG4	MLG3	MLG2	MLG1	MLG0		PLL3MULG
004E8 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP		PLL3CTRL

have the same functionality like the appropriate registers of the Main PLL. Please refer to section 14.4.2 Main PLL Control Registers (Page No.318).

14.5. PLL Interface Interrupts

All 4 PLL interfaces can generate an interrupt after a gear-up or gear-down sequence is finished. The relevant bits are in the PLLnCTRL registers:

Address	7	6	5	4	3	2	1	0	Register	
00490 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP	Main	PLLCTRL
004E0 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP	E-Ray	PLL2CTRL
004E8 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP	USB	PLL3CTRL
004F8 _H	-	-	-	-	IEDN	GRDN	IEUP	GRUP	Media LB	MPLLCTRL

The flags GRUP, GRDN are set when gear-up or gear-down, respectively. The flags can be polled by the CPU. If the interrupt enable bits IEUP or IEDN are set, an interrupt request to the CPU is generated.

All 4 interrupt request lines are OR-connected into interrupt #140 / ICR62:

Interrupt	Interrupt number		Interrupt level		Interrupt vector		DMA	
	Decimal	Hexadecimal	Setting Register	Register address	Offset	Default Vector address	RN	Stop
Timebase Overflow	140	8C	ICR62	47E _H	1CC _H	000FFDCC _H	—	—
PLL Clock Gear Data Flash Write Complete	141	8D			1C8 _H	000FFDC8 _H	— 195	— 195

14.6. Recommended Settings**14.6.1 Recommended Settings for Main PLL****Table 14.6-1 Recommended settings for Main PLL**

PLL Input (CLKMAIN) [MHz]	Multiplier Parameter		Auto-Gear Parameter		PLL Output (CLKVCO) [MHz]	CLKPLL [MHz]
	DIVM	DIVN	DIVG	MULG		
4	2	25	16	24	200	100
4	2	24	16	24	192	96
4	2	23	16	24	184	92
4	2	22	16	24	176	88
4	2	21	16	20	168	84
4	2	20	16	20	160	80
4	2	19	16	20	152	76
4	2	18	16	20	144	72
4	2	17	16	16	136	68
4	2	16	16	16	128	64
4	2	15	16	16	120	60
4	2	14	16	16	112	56
4	2	13	16	12	104	52
4	2	12	16	12	96	48
4	2	11	16	12	88	44
4	4	10	16	24	160	40
4	4	9	16	24	144	36
4	4	8	16	24	128	32
4	4	7	16	24	112	28
4	6	6	16	24	144	24
4	8	5	16	28	160	20
4	10	4	16	32	160	16
4	12	3	16	32	144	12

- Important remark: Not all settings which are shown in this table are available for all devices. Please consult the available datasheet for each device for the maximum allowed PLL output and the allowed maximum frequencies for each clock domain (CLKB, CLKP and CLKT) respectively.

14.6.2 Recommended Settings for E-Ray PLL

The clocks for E-Ray need to be set to 80 MHz. [Table 14.6-2](#) shows the recommended setting for PLL divider and clock gear.

Table 14.6-2 Recommended settings for E-Ray PLL

PLL Input CLKMAIN [MHz]	Frequency Parameter		Clockgear Parameter		PLL2 Output [MHz]	E-Ray SCLK Clock [MHz]
	PLL2DIVM	PLL2DIVN	PLL2DIVG	PLL2MULG		
4	2	20	0	0	160	80

14.6.3 Recommended Settings for USB PLL

The clock for USB needs to be set to 48 MHz. [Table 14.6-3](#) shows the recommended setting for PLL divider and clock gear.

Table 14.6-3 Recommended settings for USB PLL

PLL Input CLKMAIN [MHz]	Frequency Parameter		Clockgear Parameter		PLL3 Output [MHz]	USB Clock [MHz]
	PLL3DIVM	PLL3DIVN	PLL3DIVG	PLL3MULG		
4	2	12	0	0	96	48

14.7. Caution

14.7.1 PLL Gear-Up, Gear-Down

When using the clock auto-gear function it is strongly recommended to make use of the gear up and gear down flags (PLLxCTRL.GRUP, PLLxCTRL.GRDN) to evaluate the current state of this function to avoid malfunctions in the clock system due to setting changes prior to completion.

Please refer to section 14.5. [PLL Interface Interrupts \(Page No.324\)](#).

14.7.2 Main PLL Operation in STOP State

Basically it is allowed to keep the Main PLL running in STOP state. But the PLL should not be clock source during STOP. [Table 14.7-1](#) gives an overview:

Table 14.7-1 Main PLL in STOP state

Control signal setting during STOP				Behaviour in STOP and at wakeup
REGCTR_ MAINKPEN ¹	STCR_ OSCD1 ²	CLKR_ PLL1EN ³	CLKR_ CLKS[1:0] ⁴	
x	x	0	X0 divided by 2	recommended setting: The PLL is off in STOP. The CPU will start after wakeup with X0 clock divided by 2.
0	x	1		allowed setting: MAINKPEN=0 switches off the PLL in STOP. The CPU will start after wakeup with X0 clock divided by 2.
1	0	1		allowed setting: The PLL is running “in the background” and locked (because X0 clock is running). The CPU will start after wakeup with X0 clock divided by 2.
1	0	1	PLL	Not allowed: The PLL is running “in the background” and locked. The CPU will start after wakeup with PLL clock. When the clocks start at high frequency, there is an internal regulator voltage drop which may cause malfunction. There is no “clock gear-up” after wakeup!
0	0	1		Not allowed: MAINKPEN=0 switches off the PLL in STOP. The CPU is connected to the PLL. After wakeup, the PLL starts with unpredictable frequencies and may cause malfunction. Additionally, there may be the internal regulator voltage drop.

1. MAINKPEN=1 causes that the main voltage regulator keeps active in STOP. MAINKPEN=0 (default): the sub regulator is used in STOP.
2. OSCD1=1 disables the X0 (4 MHz) oscillator in STOP mode. OSCD1=0 (default) X0 keeps running.
3. PLL1EN=1 enables the main PLL.
4. CLKS[1:0] control the main clock source.

14.8. Procedure example for Main PLL

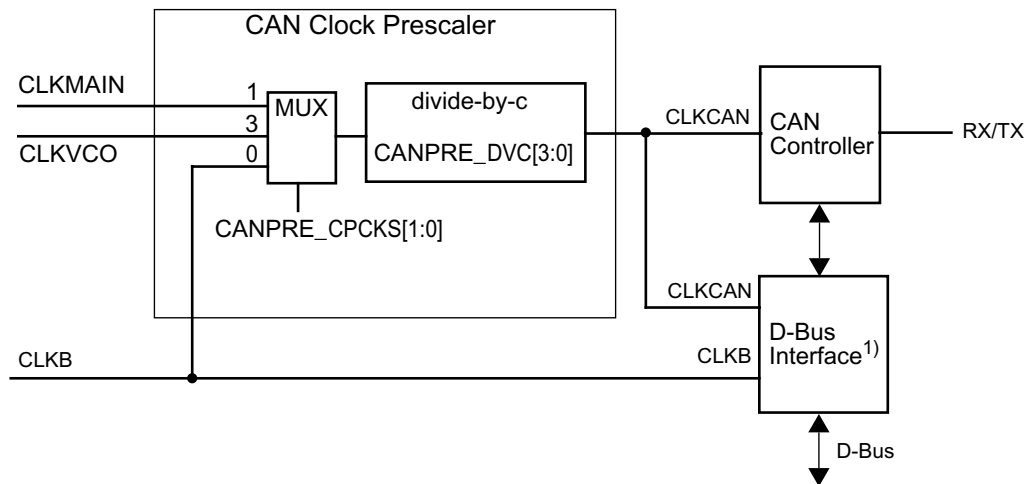
- Set the PLL interface registers (PLLDIVN, PLLDIVM, PLLDIVG, PLLMULG) according to the selected frequency and gear duration
- Switch on the PLL (CLKR.PLL1EN='1')
- If interrupts should be received after gearing up or down, also enable the corresponding interrupt enables (PLLCTRL.IEUP, PLLCTRL.IEDN)
- Wait for the PLL stabilization time
- Set the base clock division registers (DIVR0, DIVR1)
- Switch the clock source to the PLL (CLKR.CLKS "00"-> "10")
- Wait for the PLLCTRL.GRUP gear up flag (either by polling or by interrupt) before switching the clock source back to oscillation or confirm the setting of PLLCTRL.GRUP='1' before changing bits in the CLKR register
- Switch the clock source to Oscillator (CLKR.CLKS "10"-> "00")
- Wait for PLLCTRL.GRDN gear down flag (either by polling or by interrupt) before switching the clock source back to PLL or confirm the setting of PLLCTRL.GRDN='1' before changing bits in the CLKR register
- Switch off the PLL (CLKR.PLL1EN='0')

Chapter 15 CAN Clock Prescaler

15.1. Overview

- This block diagram (simplified) shows the integration of the CAN Controller and the D-Bus Interface with the CAN clock prescaler logic (divide-by-c) and clock source selector.

Figure 15.1-1 Combination of CAN Clock prescaler and CAN interface



¹⁾The D-Bus Interface is an interface between CLKB domain (D-BUS) and CLKCAN domain (CAN Controller)

- Remark:** If the CLKCAN source is set either to main oscillator or to PLL output then the clock for the CAN is not influenced by the clock modulation. If the CLKCAN source is set core clock CLKB then the clock for the CAN is also modulated (if the clock modulator is enabled).

15.2. Features

- CAN clock source selectable out of CLKMAIN, CPU Clock (CLKB) and PLL output
- Free programmable divide-by-c counter in the range of 1..16
- Individual clock disable function for each CAN controller

15.3. Registers

15.3.1 CAN Clock Control Register

Controls the CAN clock source, the clock division ratio and the clock disable.

- **CANPRE: Address 04C0h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	CPCKS1	CPCKS0	DVC3	DVC2	DVC1	DVC0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	X	X	X	X	X	X	Initial value (Software reset)
R0/W0	R0/W0	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-6: Reserved bitAlways write “0” to these register bits.
- Bit5-4: CAN Prescaler Clock Selection

CPCKS1-CPCKS0	Prescaler clock source
00	CLKB, core clock (initial)
01	CLKVCO
10	reserved
11	CLKMAIN

- Bit3-0: Source clock Divide-by-C selection

DVC3-DVC0	Source clock divided-by-C (generates CLKCAN)
0000	Source clock : 1 (no division)
0001	Source clock : 2 (division by 2)
0010	Source clock : 3 (division by 3)
0011	Source clock : 4 (division by 4)
0100	Source clock : 5 (division by 5)
0101	Source clock : 6 (division by 6)
0110	Source clock : 7 (division by 7)
0111	Source clock : 8 (division by 8)
.....
1111	Source clock : 16 (division by 16)

Note: Do not exceed the specified upper frequency limit of CLKCAN (e.g. 20MHz) e.g. by setting prescaler values exceeding this limit, or by switching the prescaler clock source to a higher frequency clock without switching previously the prescaler values to higher division rates.

Note: If prescaler source is selected to CLKVCO: Even though it is possible to select no division ratio (:1) for the divide-by-C counter it is not recommended. The resulting output clock will have an odd clock duty ratio (direct PLL output can have up to 90:10 duty). Always select at least a division ratio > 1.

MB91460 Series• **CANCKD: Address 04C1h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	CANCKD 5	CANCKD 4	CANCKD 3	CANCKD 2	CANCKD 1	CANCKD 0	
0	0	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
0	0	0	0	X	X	X	X	Initial value (Software reset)
R/W0	R/W0	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-6: Reserved bitAlways write “0” to these register bits.
- Bit5-0: CLKCAN disable

CANCKD5	Function
0	CAN controller 5 is enabled
1	CAN controller 5 is disabled

CANCKD4	Function
0	CAN controller 4 is enabled
1	CAN controller 4 is disabled

CANCKD3	Function
0	CAN controller 3 is enabled
1	CAN controller 3 is disabled

CANCKD2	Function
0	CAN controller 2 is enabled
1	CAN controller 2 is disabled

CANCKD1	Function
0	CAN controller 1 is enabled
1	CAN controller 1 is disabled

CANCKD0	Function
0	CAN controller 0 is enabled
1	CAN controller 0 is disabled

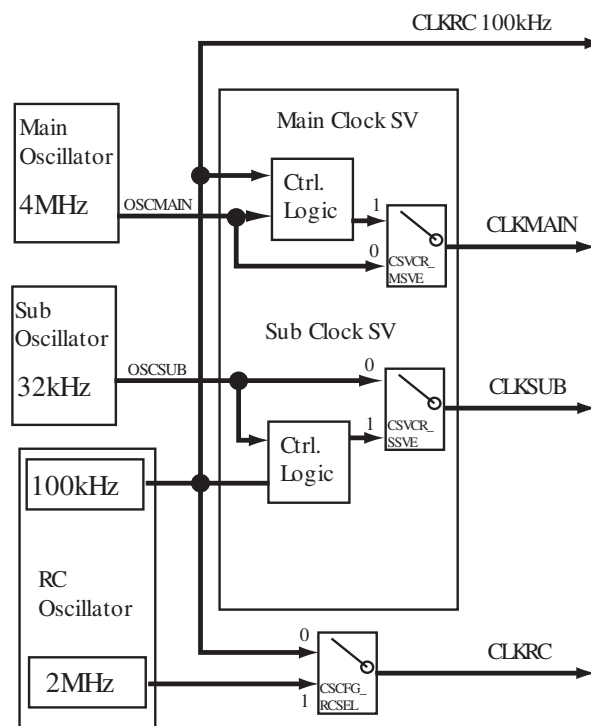
Chapter 16 Clock Supervisor

This section gives an overview of the Clock Supervisor. Purpose of the Clock Supervisor is the supervision of the Main- and Sub oscillators. In case of oscillation (OSCMAN or OSCSUB) failure the Clock Supervisor control logic will take action, i.e. switching to an internal RC-oscillation clock (CLKRC 100kHz), depending on the operation mode set in the control register.

In MB91FV460B, MB91F467P and later devices, an new Clock Supervisor version with extended functionality is implemented. This new feature is marked with the keyword “New feature”.

16.1. Overview Clock Supervisor

Figure 16.1-1 Block diagram of the clock supervisor



The purpose of the clock supervisor is the supervision of the main and sub oscillation clocks. In case of a oscillation failure (OSCMAN and/or OSCSUB) it can be replaced by an on-chip RC-oscillation clock (CLKRC 100kHz), depending on the configuration.

If a clock the MCU currently uses, fails for a certain time (20-80 μ s for *Main* clock / 160-640 μ s for Sub clock) the MCU is reset by Setting Initialization Request (INIT) and the reset cause can be checked after reset vector fetch. If the Sub clock is failing while the MCU is in Main clock mode, reset will be delayed until the transition to Sub clock mode or no reset will be initiated. The user can choose the behaviour with a control bit in the Clock Supervisor Control Register.

There are two independent supervisors, one for the Main clock and one for the Sub clock. They can be enabled/disabled separately.

Main clock and Sub clock supervisor are disabled and re-enabled automatically if the corresponding oscillator is disabled and re-enabled.

If the MCU changes to STOP state, the RC-oscillator can be automatically disabled by a control bit. It will be enabled again upon wake-up from STOP state.

There are two status bits in the Clock Supervisor Control Register which indicate the failure of the Main clock and Sub clock. These bits can be available at two port pins (device dependent). Single clock devices can use the CLKRC as Sub clock.

New feature: The two Clock Supervisor status bits can be cleared by CPU access, if the main and/or sub oscillator has resumed oscillation. The clock is switched back to OSCMAIN and/or OSCSUB in this case.

New feature: The RC oscillator is enabled in STOP mode automatically, if the Hardware Watchdog is configured to run during STOP. The RC oscillator can **only** be stopped in STOP mode, and then it depends on the Hardware Watchdog and the control bit in the Clock Supervisor Control Register.

16.2. Clock Supervisor Register

This section lists the Clock Supervisor Control Register and describes the function of each bit in detail.

■ Clock Supervisor Control Register (CSVCR)

The Clock Supervisor Control Register (CSVCR) sets the operation mode of the Clock Supervisor. Figure 16.2-1 shows the configuration of the Clock Supervisor Control Register.

Figure 16.2-1 Configuration Clock Supervisor Control Register (CSVCR)

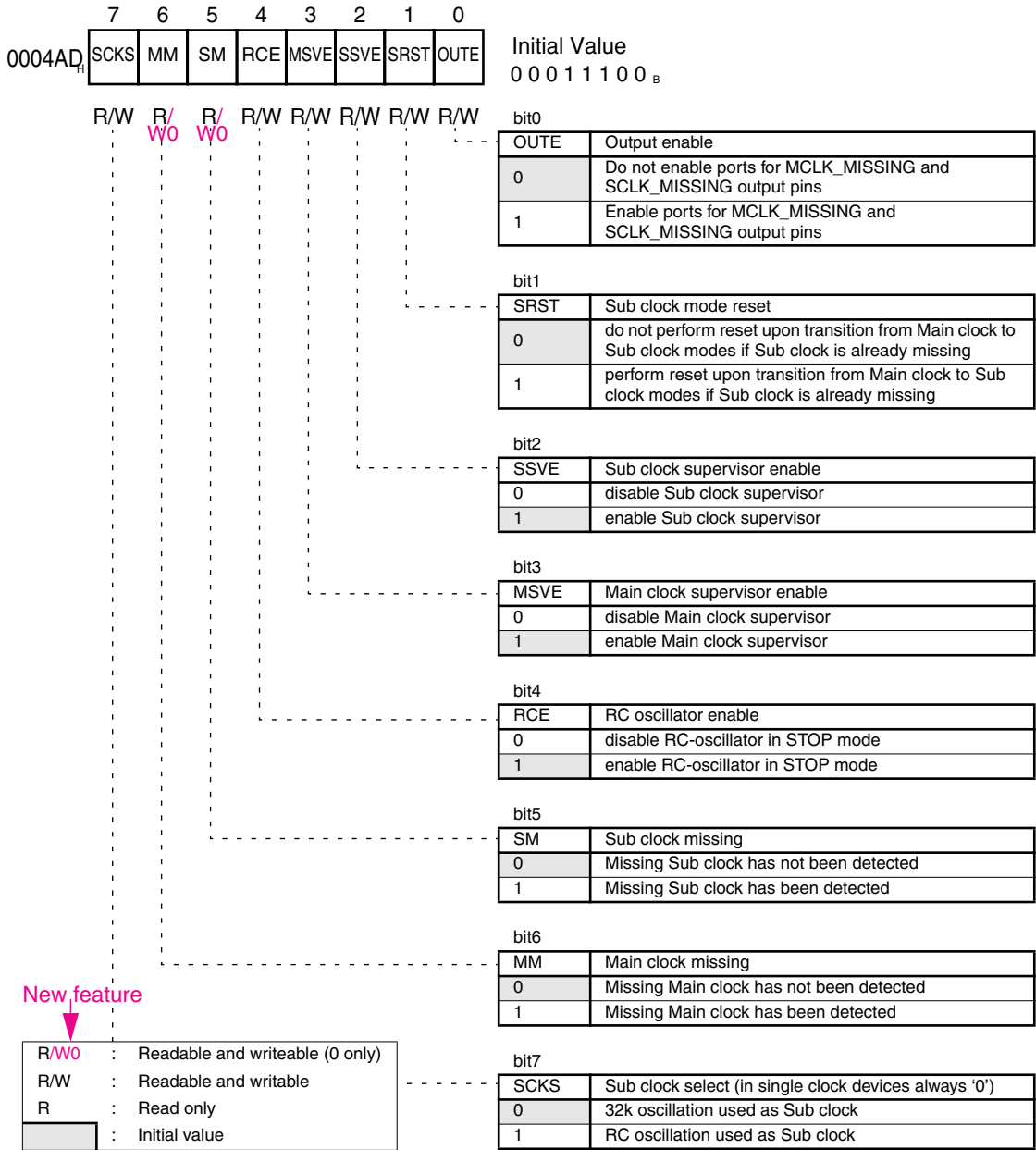


Table 16.2-1 describes the function of each bit of the Clock Supervisor Control Register (CSVCR).

Table 16.2-1 Functional Description of each bit of the Clock Supervisor Control Register

Bit	Name	Function
7	SCKS (Sub clock select)	This bit is to select between 32 kHz external oscillation and internal RC oscillation as Sub clock. If this bit is '0' then the external 32 kHz oscillation is used as Sub clock, if it's '1' then the internal RC oscillation is used as Sub clock. This bit is cleared to '0' by Power-On reset or external reset. Other types of reset will not affect this bit. Modifying only in main clock mode allowed!
6	MM (Main clock missing)	If this bit is 1, the Main clock supervisor has detected that the Main oscillation clock coming from X0, X1 is missing, e.g. by a broken crystal. If this bit is '0', a missing Main clock has not been detected. This bit is cleared to '0' by Power-On reset or external reset. Other types of reset will not affect this bit. New feature: This bit can be cleared by CPU access, if the main oscillator has resumed oscillation. If the main oscillator is still failing, the write access is ignored.
5	SM (Sub clock missing)	If this bit is 1, the Sub clock supervisor has detected that the sub oscillation clock coming from X0A, X1A is missing, e.g. by a broken crystal. If this bit is '0', a missing Sub clock has not been detected. This bit is cleared to '0' by Power-On reset or external reset. Other types of reset will not affect this bit. New feature: This bit can be cleared by CPU access, if the sub oscillator has resumed oscillation. If the sub oscillator is still failing, the write access is ignored.
4	RCE (RC-oscillator enable)	Setting this bit to '1' enables the RC-oscillator in STOP mode. Outside STOP mode, the RC-oscillator is always enabled. This bit is set to '1' by Power-On reset or external reset. Other types of reset will not affect this bit. New feature: If HWWD.STP_RUN (=HWWD[4]) is set in the Hardware Watchdog, then the RC oscillator is enabled and read and read-modify-write operations will return '1' independently of RCE register setting. Effective RCE = RCE_Register or HWWD.STP_RUN
3	MSVE (Main clock supervisor enable)	Setting this bit to '1' enables the Main clock supervisor. This bit is set to '1' by Power-On reset only. Other types of reset will not affect this bit.
2	SSVE (Sub clock supervisor enable)	Setting this bit to '1' enables the Sub clock supervisor. This bit is set to '1' by Power-On reset only. Other types of reset will not affect this bit.
1	SRST (Sub clock mode reset)	If this bit is set to '1', a reset is performed upon transition from Main/PLL clock mode to Sub clock mode if the Sub clock is already missing. If this bit is set to '0', no reset is performed in this case. This bit is cleared to '0' by Power-On reset or external reset. Other types of reset will not affect this bit.
0	OUTE (Output enable)	This bit can be used as an output enable to output the signals MCLK_MISSING (bit 3 of CSVCR) and SCLK_MISSING (bit 4 of CSVCR) to port pins. For more information about the pins see the corresponding Datasheet. If this bit is set to '1', the ports are enabled for MCLK_MISSING and SCLK_MISSING output. This bit is cleared to '0' by Power-On reset or external reset. Other types of reset will not affect this bit.

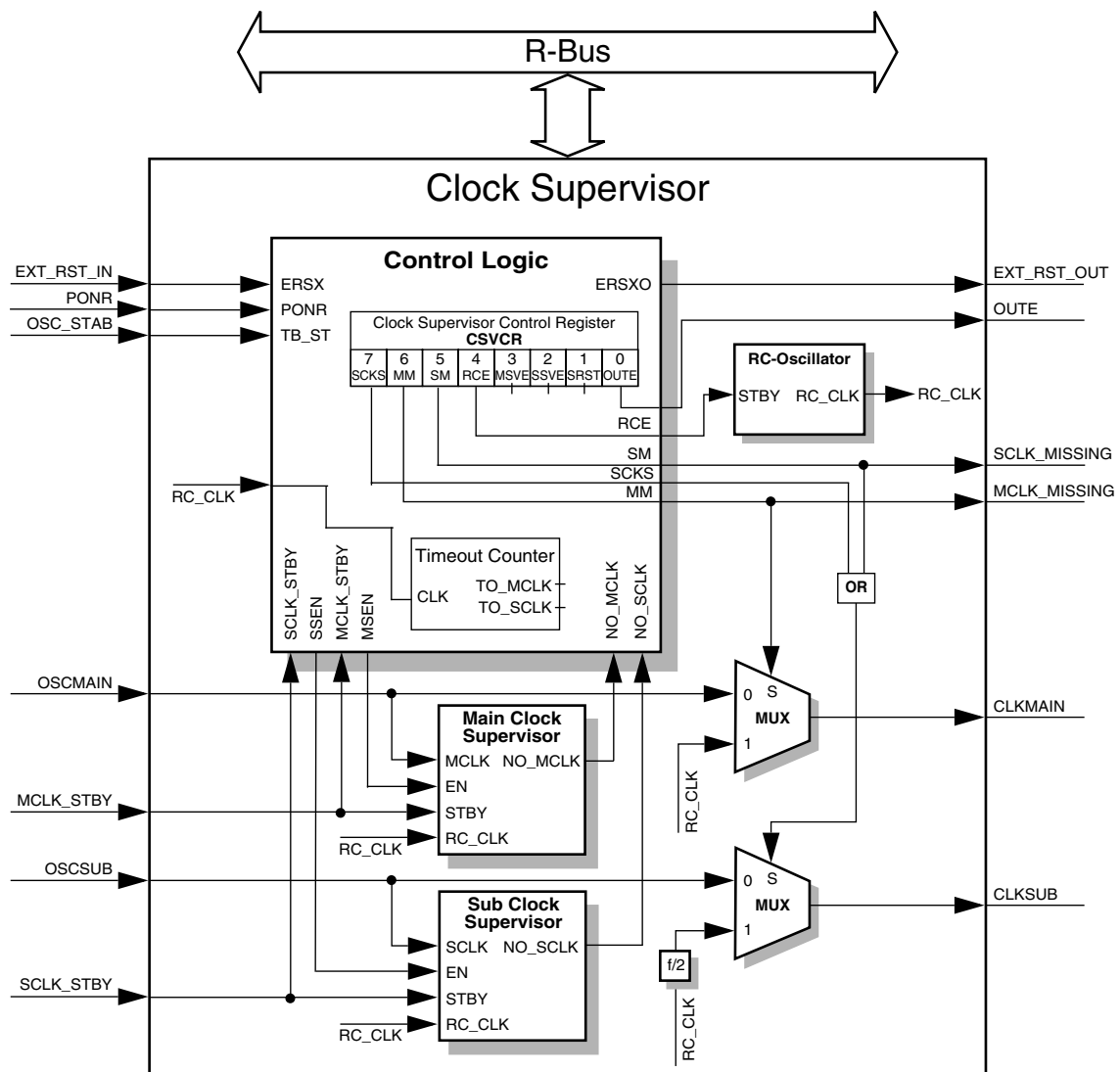
16.3. Block Diagram Clock Supervisor

This section presents a block diagram of the Clock Supervisor. The building blocks of the Clock Supervisor are:

- | Main Clock Supervisor
- | Sub Clock Supervisor
- | Control Logic
- | RC-Oscillator

■ Block Diagram Clock Supervisor

Figure 16.3-1 Block Diagram of Clock Supervisor



SCLK_OUT and MCLK_OUT can be observed using the Clock Monitor Module. SCLK_MISSING and MCLK_MISSING can be programmed to device specific outputs (see the datasheet of the used device for the information which pins are used) by setting OUTE=1.

Signal EXT_RST_IN is the reset input, connected to the external INITX pin.

Signal EXT_RST_OUT is the reset output and causes Setting Initialization Request (INIT).

16.4. Operation Modes

This section describes all operation modes of the Clock Supervisor.

■ Operation mode with initial settings

In case the clock supervisor control register (CSVCR) is not configured at the beginning of the user program, the RC-oscillator, the Main clock supervisor and the Sub clock supervisor is enabled.

- The RC-oscillator is enabled at power-on.
- The Main clock supervisor is enabled after the 'oscillation stabilization wait time' or in case the Main clock is missing before the completion of the 'oscillation stabilization wait time', after the 'Main clock timeout' (TO_MCLK, 20-80ms for Main clock) from the timeout counter. That means: Main clock supervisor becomes enabled if either 'Main clock timeout' or 'oscillation stabilization wait time' is reached. Whichever occurs first. The timeout counter is clocked with CLKRC. If the Main clock is missing from power-on, the power-on reset state is never left, which in this case is a safe state. The user must make sure with external pull-up/pull-down resistors that all relevant signal are pulled to the correct level.
- The Sub clock supervisor is enabled after the completion of the 'Sub clock timeout' (TO_SCLK, 1.3-5.25s for Sub clock) from the timeout counter. The timeout counter is clocked with CLKRC.
- If the Main clock stops while the Main clock supervisor is enabled, the Main clock is replaced with CLKRC 100kHz, the MM bit is set to '1' and reset (EXT_RST_OUT) is asserted.
- If the Sub clock stops and the Sub clock supervisor is enabled, the behaviour depend on whether the MCU is in Main clock mode or in Sub clock mode. If the Sub clock stops in Sub clock mode, CLKRC divided by two substitutes the Sub clock, the SM bit is set to '1' and reset (EXT_RST_OUT) is asserted. If the Sub clock stops in Main clock mode, CLKRC divided by two substitutes the Sub clock, the SM bit is set to '1' and no reset occurs upon transition to Sub clock mode, since the SRST bit has its initial value of '0'. If the SRST bit is '1' a reset (INIT) occurs.

Figure 16.4-1 Timing Diagram: Initial settings, Main clock missing during power-on reset

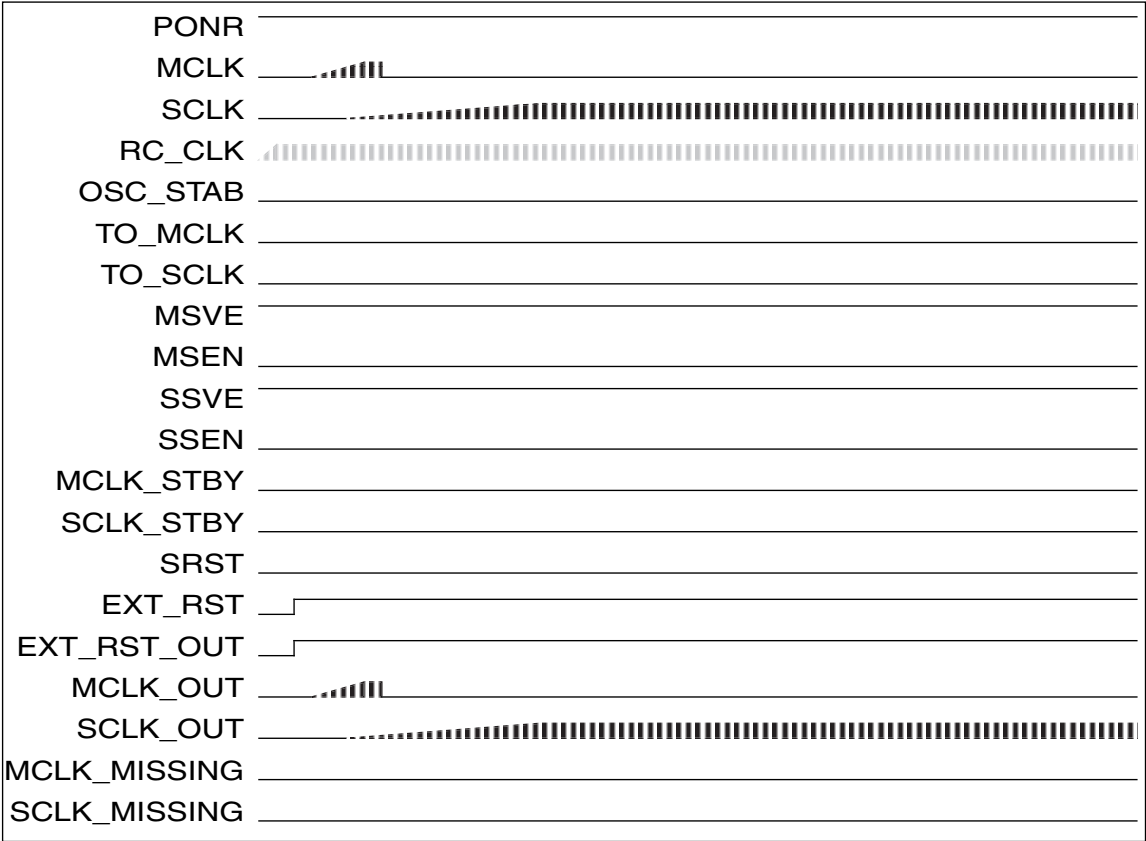


Figure 16.4-2 Timing Diagram: Initial settings, Main clock missing during 'oscillation stabilization wait time'

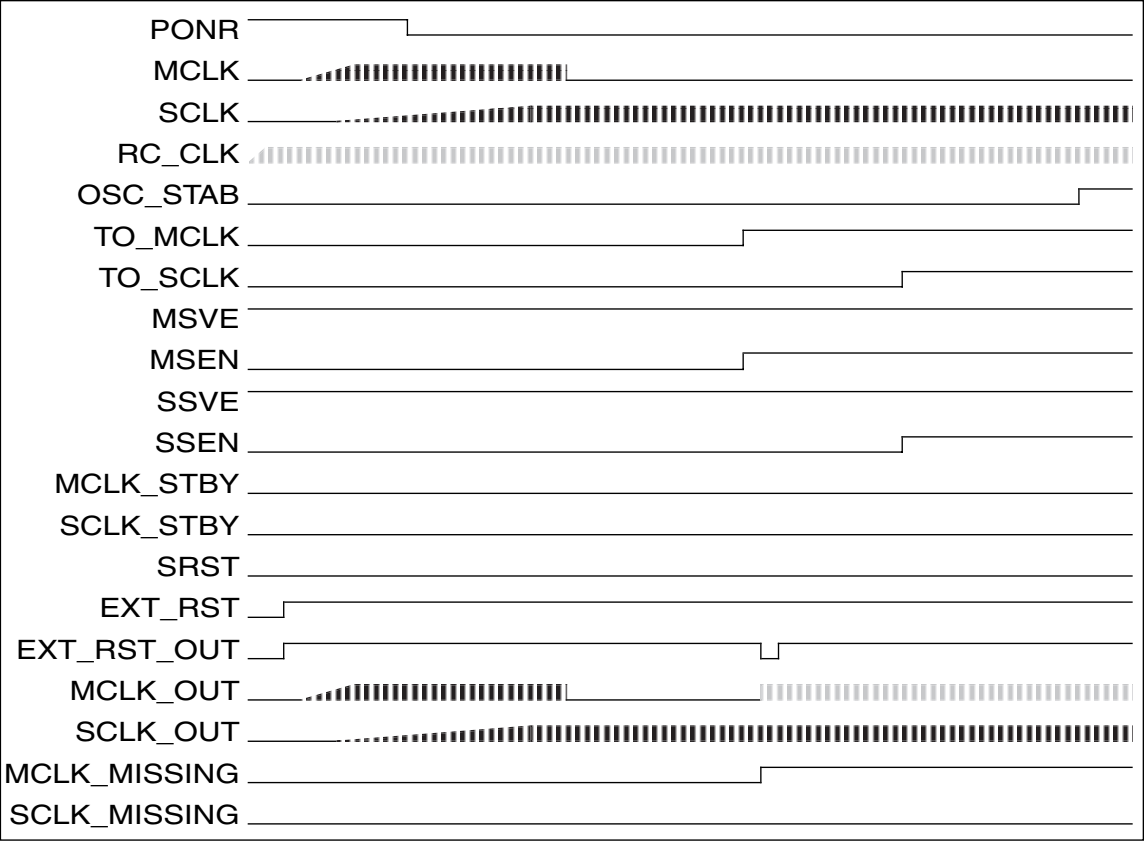


Figure 16.4-3 Timing Diagram: Initial settings, Main clock missing after 'oscillation stabilization wait time'

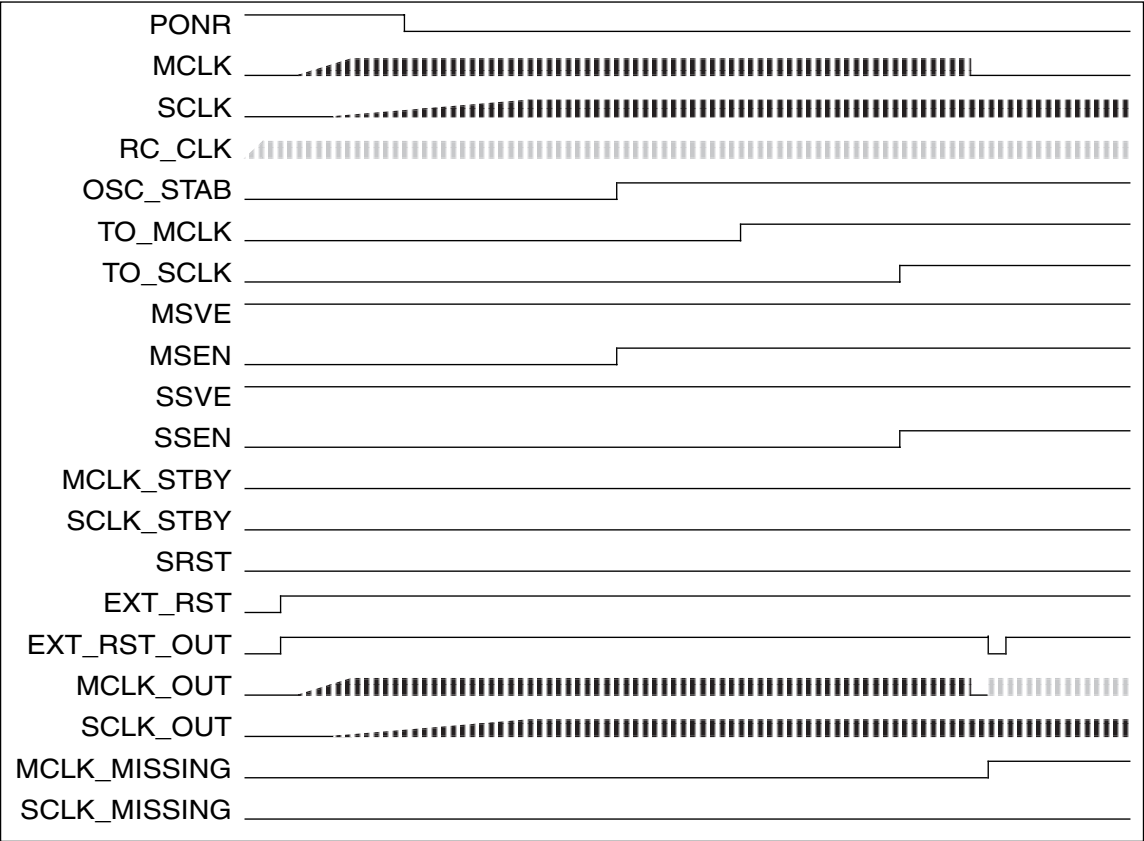


Figure 16.4-4 Timing Diagram: Initial settings, Sub clock missing before timeout

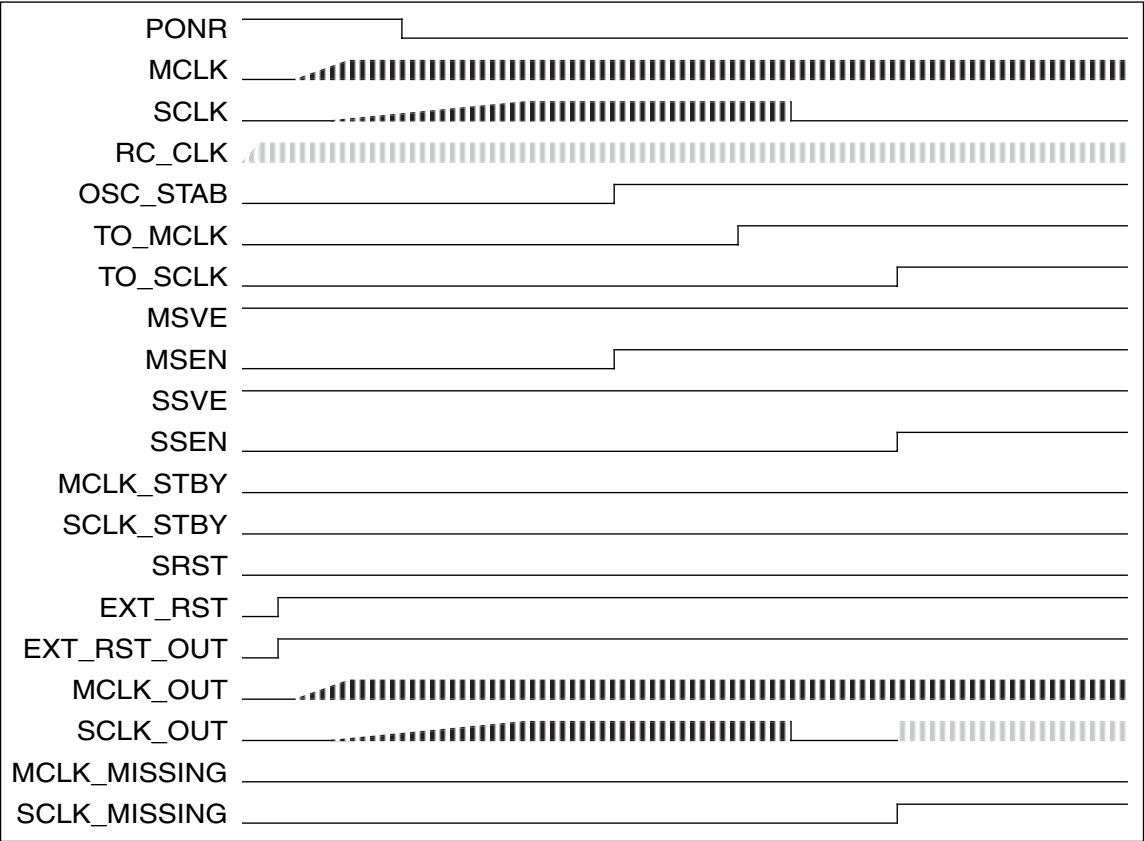
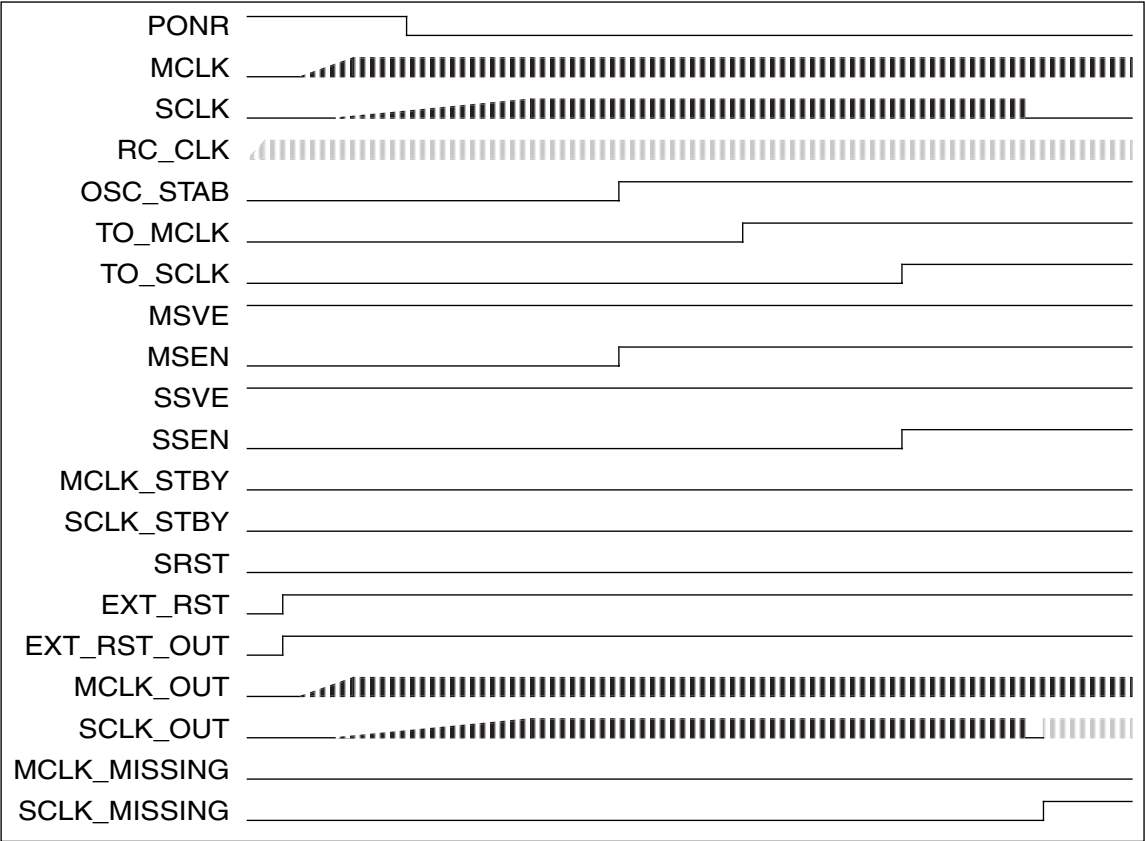


Figure 16.4-5 Timing Diagram: Initial settings, Sub clock missing after timeout

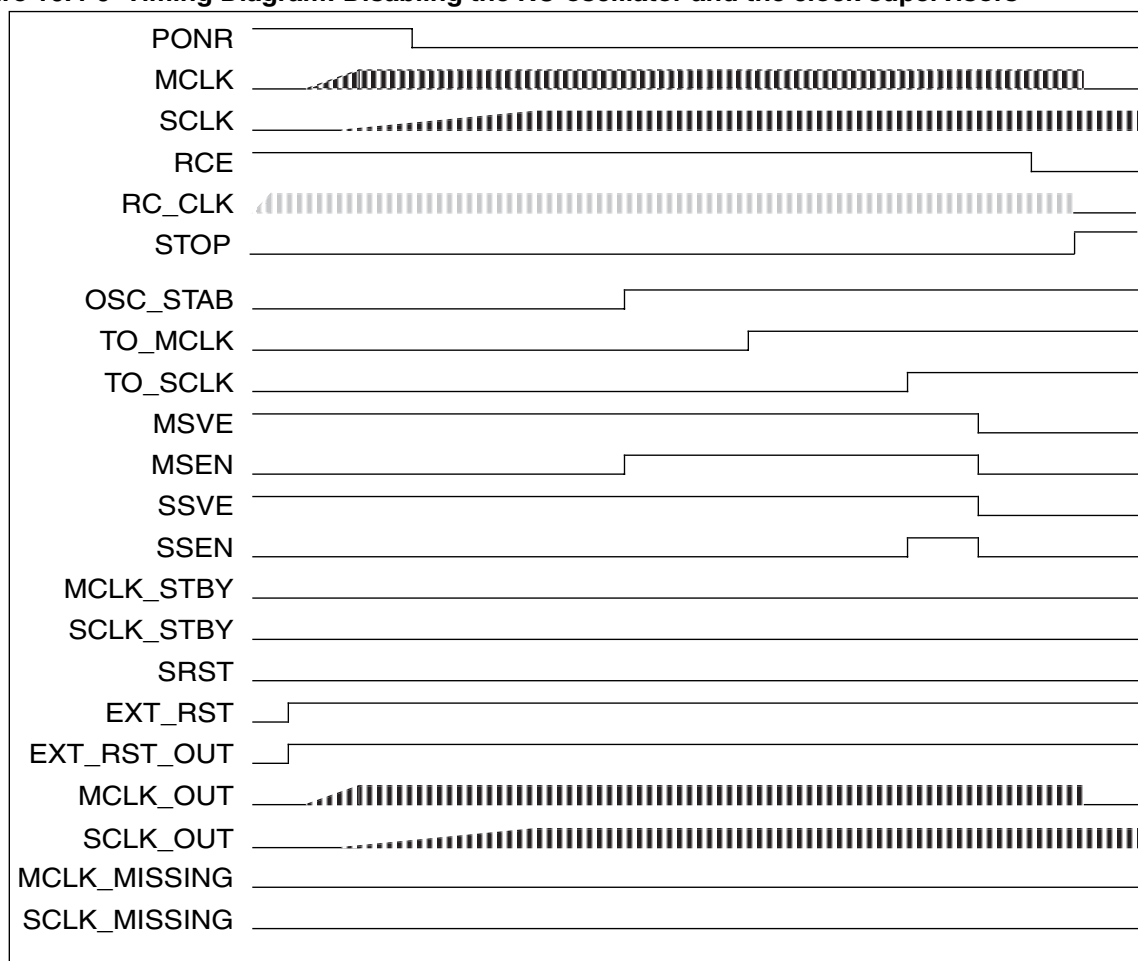


■ Disabling the RC-oscillator and the clock supervisors

The initial point of this scenario is that the RC-oscillator and Main clock or Sub clock supervisor is enabled.

- The RC-oscillator can be disabled only in STOP mode.
First check that both SM and MM (bit 5 and bit 6 of CSVCR) are '0'.
Then disable the RC-oscillator by setting RCE to '0'. If either SM or MM bit is '1', RCE must not be set to '0'!
- **New feature:** If the Hardware Watchdog is to run in STOP mode (HWWD.STP_RUN='1') then the RC-oscillator is enabled by hardware.
- The Main clock supervisor is disabled by setting MSVE (bit 3 of CSVCR) to '0'.
- The Sub clock supervisor is disabled by setting SSVE (bit 2 of CSVCR) to '0'.

Figure 16.4-6 Timing Diagram: Disabling the RC-oscillator and the clock supervisors

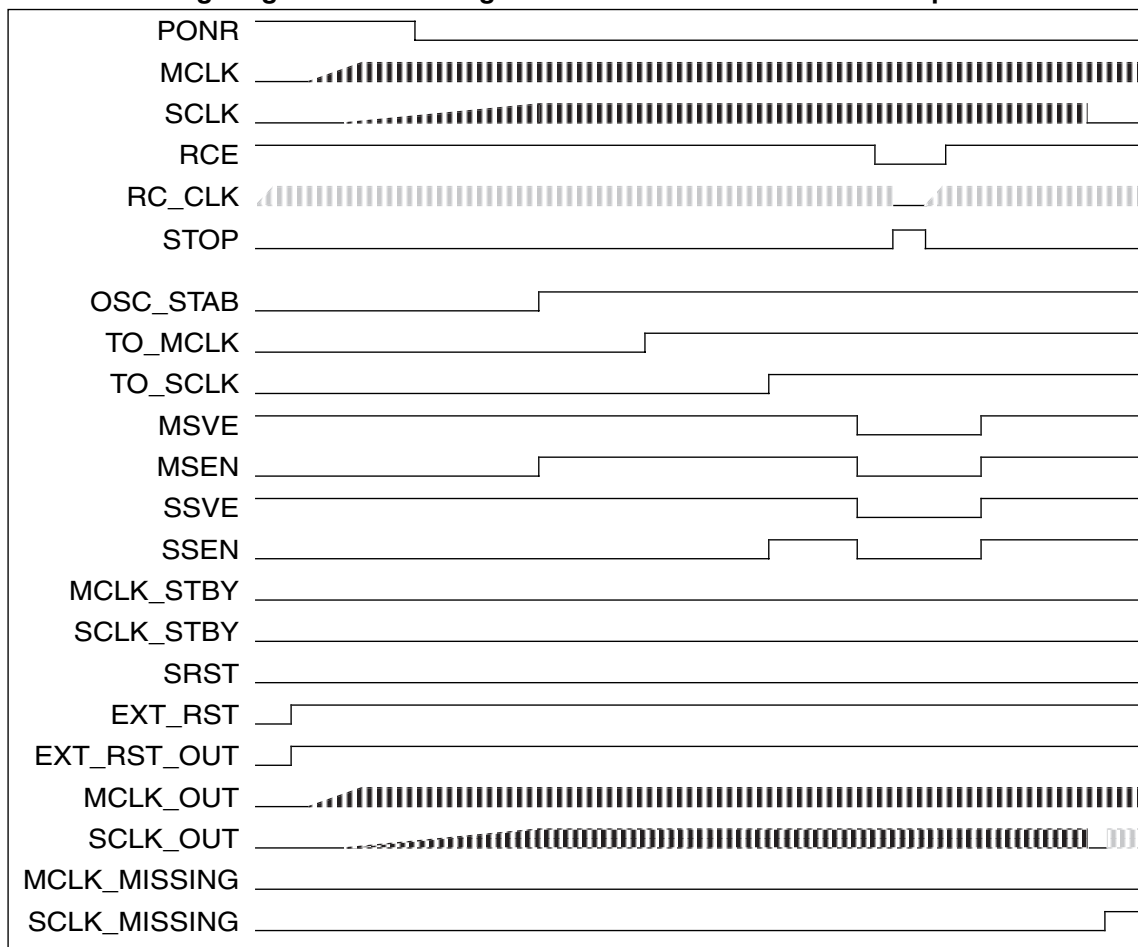


■ Re-enabling the RC-oscillator and the clock supervisors

The initial point of this scenario is that the RC-oscillator and both Main clock and Sub clock supervisor are disabled.

- The RC-oscillator is always enabled in RUN state. It can only be disabled in STOP, and after wakeup from STOP it will re-start automatically.
- The Main clock supervisor is enabled by setting MSVE (bit 3 of CSVCR) to '1'.
- The Sub clock supervisor is enabled by setting SSVE (bit 2 of CSVCR) to '1'.

Figure 16.4-7 Timing Diagram: Re-enabling the RC-oscillator and the clock supervisors



■ New feature: Switching back from RC to Main Oscillation

The initial point of this scenario is that the Main clock was missing, the Main clock supervisor has set the MM flag and switched to RC clock. The CPU already got reset (INIT) from clock supervisor and has detected MM=1 as reset source (See "Check if reset was asserted by the Clock Supervisor" on P. 354). The user is quite sure that the Main clock returned meanwhile or will return soon and wants to switch back to Main clock.

- The MM flag can be cleared by writing '0' (bit 6 of CSVCR).
- If the Main clock is still missing during the write access, the write operation has no effect, the MM flag keeps '1' value and the clock supervisor continues giving out RC clock.
- If the Main clock is operating during the write access, the MM flag is cleared and the clock is switched back to Main clock.
- It is possible to poll the MM flag until the Main clock is resumed:

```
ldi      #_csvcr,r1
clear_CSV_loop:
bandh    #0b1001,@r1      ;; Clear MM+SM
btsth    #0b0110,@r1      ;; Check: Is one of them 1?
bne      clear_CSV_loop
```

■ New feature: Switching back from RC to Sub Oscillation

The initial point of this scenario is that the CPU is running on Sub clock and Sub clock was missing. The Sub clock supervisor has set the SM flag and switched to RC clock (divided by 2). A clock supervisor reset was not generated because of CSVCR.SRST was '0'. Now the CPU is running user software on RC clock. The flag SM=1 was found by polling. The user is quite sure that the Sub oscillation returned meanwhile or will return soon and wants to switch back to Sub oscillation.

- The SM flag can be cleared by writing '0' (bit 5 of CSVCR).
- If the Sub clock is still missing during the write access, the write operation has no effect, the SM flag keeps '1' value and the clock supervisor continues giving out RC clock.
- If the Sub clock is operating during the write access, the SM flag is cleared and the clock is switched back to Sub clock.
- It is possible to poll the SM flag like described in the Main clock example above.

■ Sub clock modes

The Main clock supervisor can be configured to be turned off in Sub clock modes automatically (See "How to halt the Main clock in Sub clock mode" on P. 310). In this case the enable bit MSVE remains unchanged. At transition from Sub clock mode to Main clock mode the Main clock supervisor is enabled after the 'oscillation stabilization wait time' or in case the Main clock is missing before the completion of the 'oscillation stabilization wait time', after the 'Main clock timeout' (TO_MCLK) from the timeout counter. The timeout counter is clocked with CLKRC.

■ Changing the behaviour upon transition to Sub clock mode if the Sub clock has already stopped in Main clock mode

If the Sub clock has stopped in Main clock mode and this was detected by the Sub clock supervisor, the behaviour upon transition to Sub clock mode depends on the state of the SRST bit.

- If SRST is set to '0' (initial value), reset is not asserted at the transition to Sub clock mode. The transition is performed using the RC-oscillation clock as Sub clock. In this case it is recommended to check the SM bit before the transition to Sub clock mode to get the information if Sub clock or CLKRC is used.
- If SRST is set to '1', reset is asserted at the transition to Sub clock mode.

The following timing diagrams (Figure 16.4-8, Figure 16.4-9, Figure 16.4-10) illustrate this behaviour.

Figure 16.4-8 Timing Diagram: Sub clock missing in Main clock mode, SRST=0

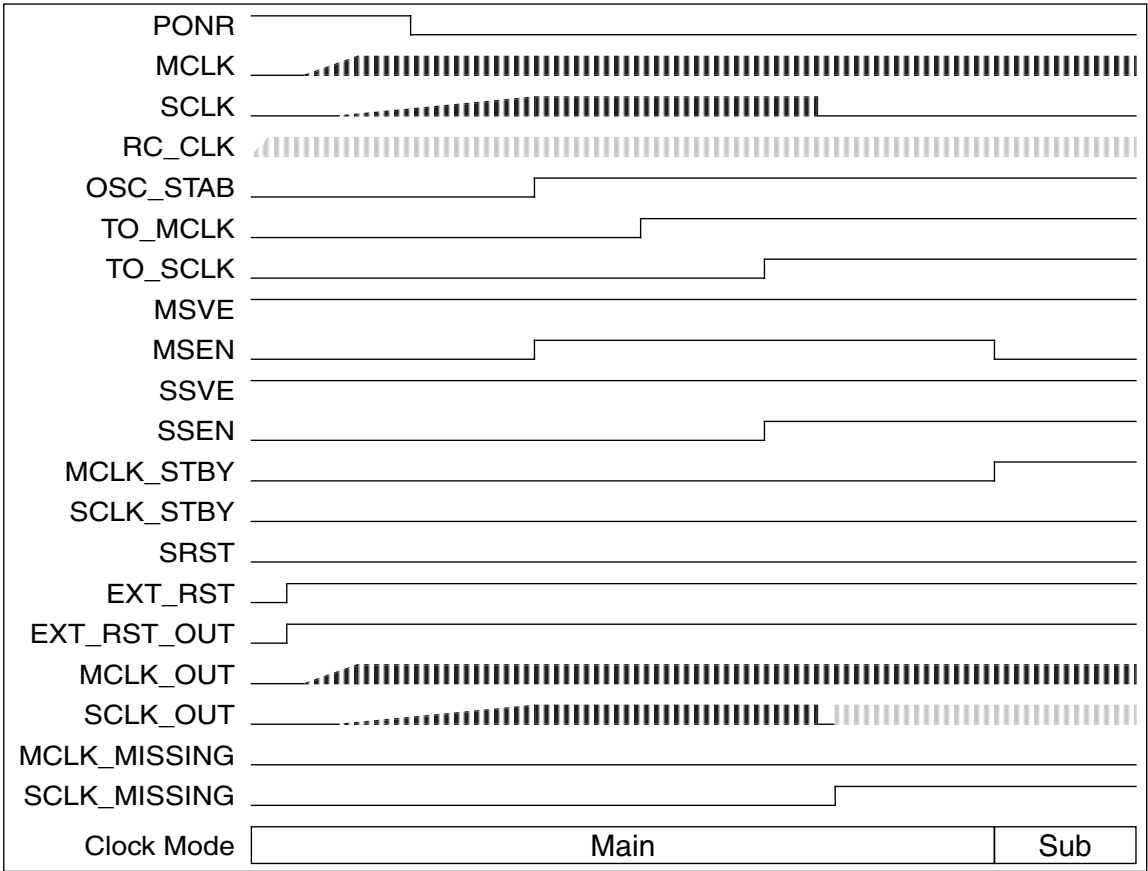


Figure 16.4-9 Timing Diagram: Sub clock missing in Main clock mode, SRST=1

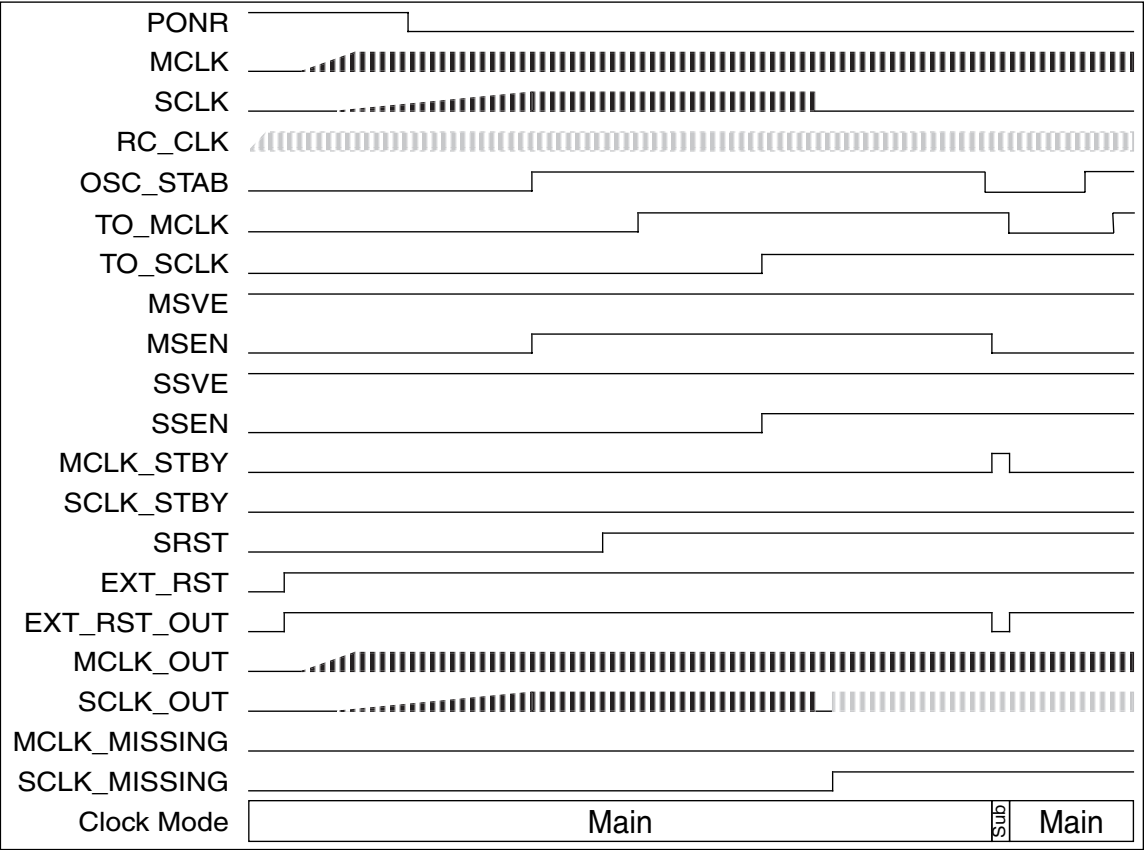
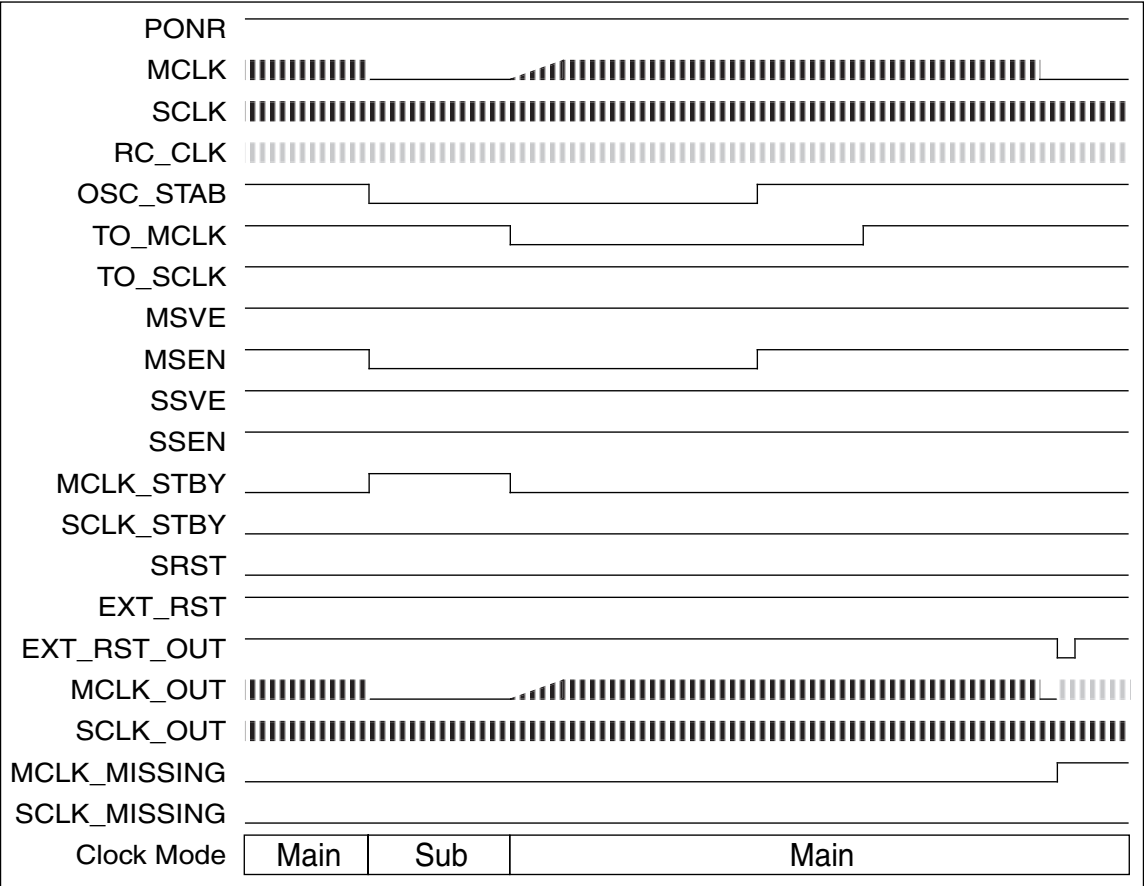


Figure 16.4-10 Timing Diagram: Waking up from Sub clock mode



■ STOP mode (with both oscillators disabled)

In this section, “STOP mode” means that the CPU is in STOP state and both oscillators are disabled by setting `STCR.OSCD1=’1’` and `STCR.OSCD2=’1’`. The Clock Supervisor’s inputs `MCLK_SBY` and `SCLK_SBY` are connected to the oscillator disable lines `OCSD1` and `OCSD2`, respectively.

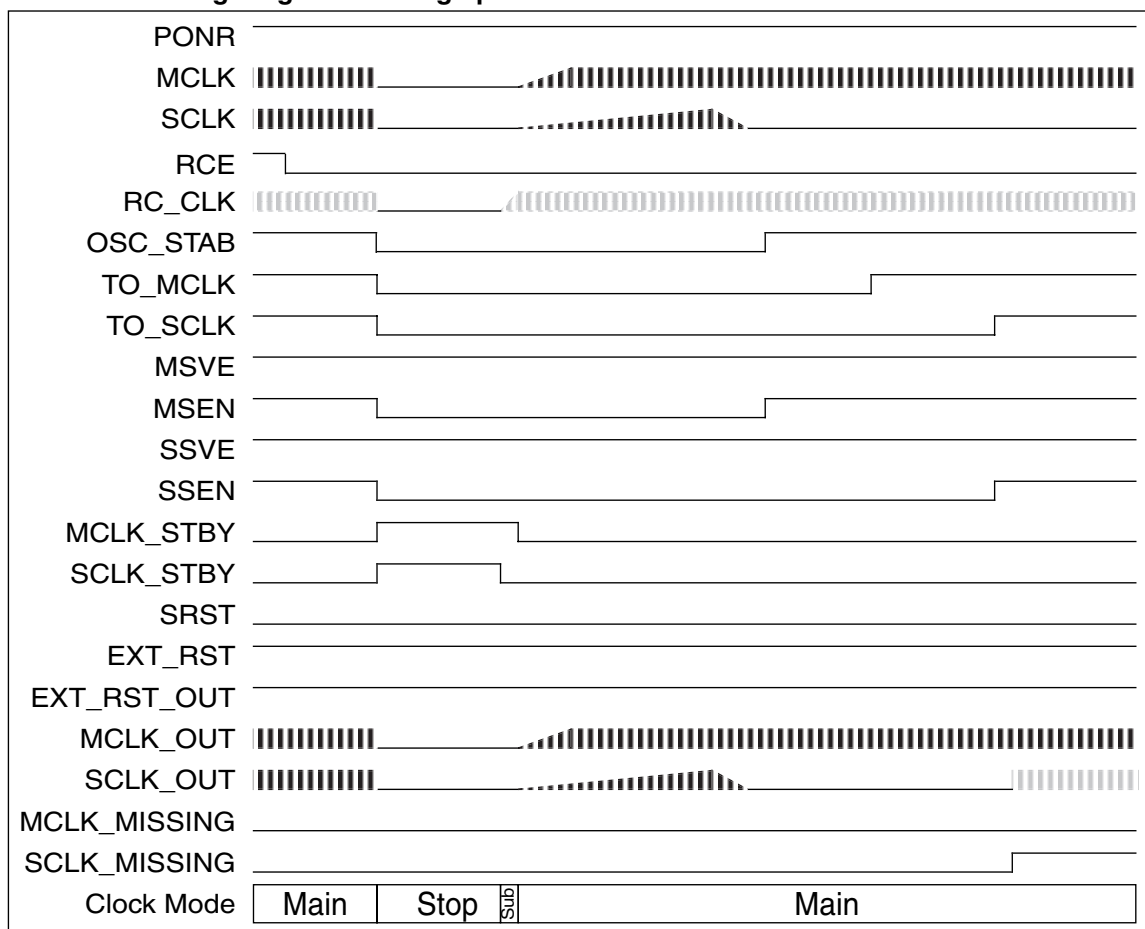
If Main clock and Sub clock supervisors are enabled, they will be automatically disabled at transition into STOP state. The corresponding enable bits in the clock supervisor control register remain unchanged. So after wake-up from STOP mode the clock supervisors will be enabled again. If the corresponding enable bits are set to ‘0’, the clock supervisors will stay disabled after wake-up from STOP mode.

The RC-oscillator is disabled in STOP, if the `RCE` bit in the `CSVCR` register is cleared.

New feature: If the Hardware Watchdog is enabled in STOP state (`HWWD.STP_RUN=’1’`), then the RC-oscillator is enabled by hardware during STOP. The `RCE` bit is unchanged, but read and read-modify-write operations return ‘1’.

- The RC-oscillator is enabled immediately after wake-up from STOP mode.
- The Main clock supervisor is enabled after the ‘oscillation stabilization wait time’ or in case the Main clock is missing after wake-up from STOP mode, after the ‘Main clock timeout’ (`TO_MCLK`) from the timeout counter. The timeout counter is clocked with `CLKRC`.
- The Sub clock supervisor is enabled after the ‘Sub clock timeout’ (`TO_SCLK`) from the timeout counter which is clocked with the `CLKRC`.

Figure 16.4-11 Timing Diagram: Waking up from STOP state



MB91460 Series**■ RTC mode (STOP mode with Real Time Clock enabled)**

In this section, “RTC mode” means that the CPU is in STOP state and one of the quartz oscillators is enabled by setting STCR.OSCD1=‘0’ or STCR.OSCD2=‘0’. The enabled oscillator clock is switched to the Real Time Clock to keep it running during STOP. The behaviour of the Clock Supervisor depends on several settings.

- If the RTC is connected to Main clock, the behaviour of the main clock supervisor is like described in Table 16.4-1

Table 16.4-1 Main Clock Supervisor in RTC mode.

RC oscillator enable CSVCR.RCE	Main Oscillator disable STCR.OSCD1	Main clock supervisor enable SVCR.MSVE	Behaviour in STOP mode if Main clock fails and the RTC is connected to Main clock
1	1	X	Main clock fail cannot be seen because the Main oscillator is disabled. The Main clock supervisor is disabled because of the Main oscillator is disabled. The RTC will not run because of the same reason. Note: This is no RTC mode.
1	0	1	The clock supervisor will set MM flag, switch the Main clock to RC clock and generate an reset (INIT) to CPU. The STOP mode is cancelled by the reset. The RTC is initialized by the reset.
1	0	0	Main clock supervisor is disabled by MSVE=0. In case of Main clock fail, the RTC clock simply stops.
0	X	X	Main clock supervisor is disabled because of it does not get RC clock. In case of Main clock fail, the RTC clock simply stops.

Note **New feature**: RCE setting is valid if HWWD.STPRUN (HWWD[4]) is ‘0’. Otherwise, RCE is overwritten to ‘1’.

- If the RTC is connected to Sub clock, the behaviour of the sub clock supervisor is like described in Table 16.4-2

Table 16.4-2 Sub Clock Supervisor in RTC mode.

RC oscillator enable CSVCR.RCE	Sub Oscillator disable STCR.OSCD2	Sub clock supervisor enable SVCR.SSVE	Behaviour in STOP mode if Sub clock fails and the RTC is connected to Sub clock
1	1	X	Sub clock fail cannot be seen because the Sub oscillator is disabled. The Sub clock supervisor is disabled because of the Sub oscillator is disabled. The RTC will not run because of the same reason. Note: This is no RTC mode.
1	0	1	The clock supervisor will set SM flag and switch the Sub clock to RC clock. The RTC continues running on RC clock. A reset is not generated because there is no transition from Main clock to Sub clock during STOP mode.
1	0	0	Sub clock supervisor is disabled by SSVE=0. In case of Sub clock fail, the RTC clock simply stops.

Table 16.4-2 Sub Clock Supervisor in RTC mode.

RC oscillator enable CSVCR.RCE	Sub Oscillator disable STCR.OSCD2	Sub clock supervisor enable SVCR.SSVE	Behaviour in STOP mode if Sub clock fails and the RTC is connected to Sub clock
0	X	X	Sub clock supervisor is disabled because of it does not get RC clock. In case of Sub clock fail, the RTC clock simply stopps.

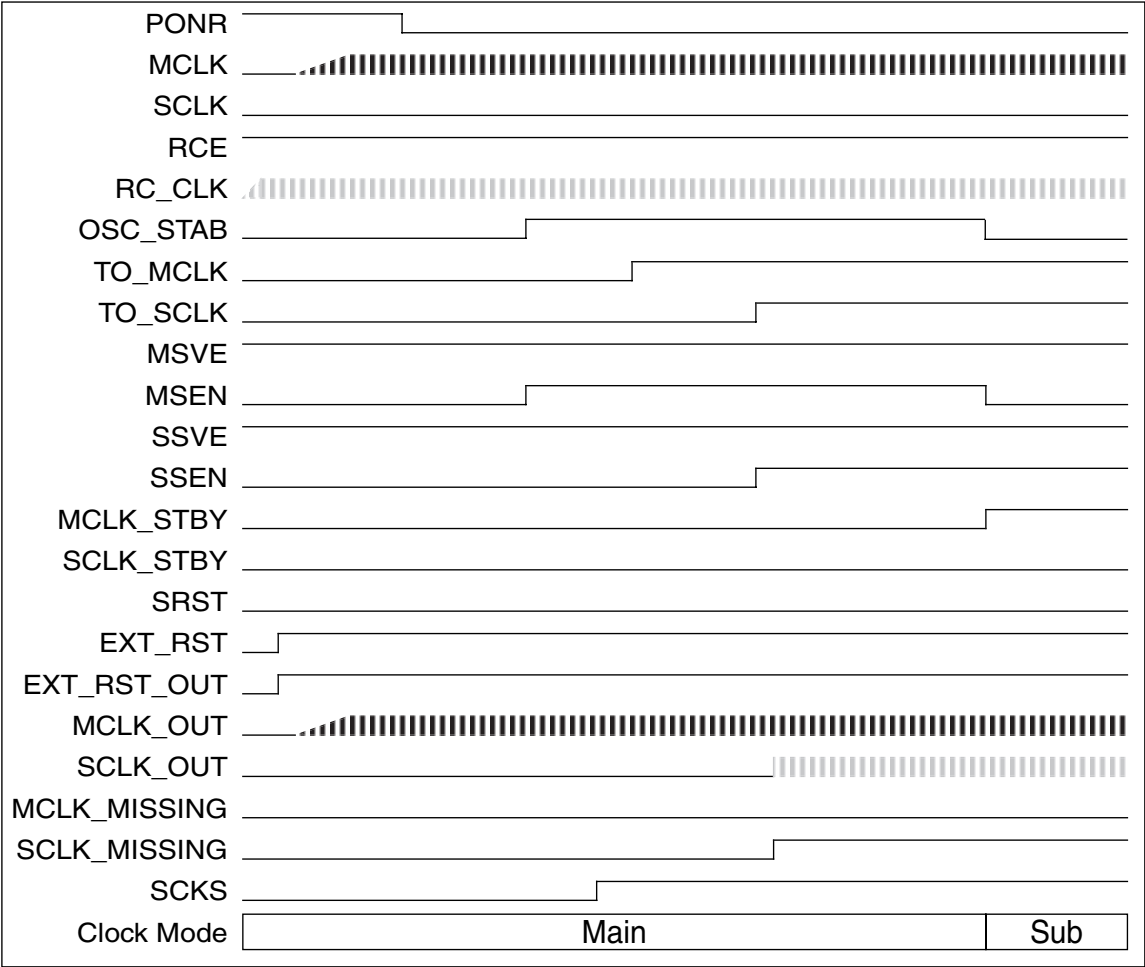
Note **New feature**: RCE setting is valid if HWWD.STPRUN (HWWD[4]) is '0'. Otherwise, RCE is overwritten to '1'.

- The RC-oscillator is enabled immediately after wake-up from STOP state.
- If the Main clock was disabled in STOP: The Main clock supervisor is enabled after the 'oscillation stabilization wait time' or in case the Main clock is missing after wake-up from STOP state, after the 'Main clock timeout' (TO_MCLK) from the timeout counter. The timeout counter is clocked with CLKRC.
- IF the Sub clock was disabled in STOP: The Sub clock supervisor is enabled after the 'Sub clock timeout' (TO_SCLK) from the timeout counter which is clocked with the CLKRC.

■ RC-Clock as Sub Clock

The Sub clock supervisor can provide the CLKRC as Sub clock. To enable this feature, SCKS bit (bit7 of CSVCR) must be set to '1'. The switching between 32k oscillation and RC oscillation for Sub clock have to take place only in main clock mode. Otherwise spikes may be generated.

Figure 16.4-12 Timing Diagram: Sub clock mode with single clock device



■ Check if reset was asserted by the Clock Supervisor

To find out whether the Clock Supervisor has asserted reset, the software must check the reset cause by reading the RSRR register (See "[RSRR: Reset Cause Register](#)" on P. 229). On the most flash devices, the RSRR register is read and cleared by the Boot ROM software. The content of RSRR can be found in CPU register R4[7:0] after Boot ROM is done.

If INIT (bit 7 of RSRR) is set, the cause was either external reset at the INITX pin or the clock supervisor or the hardware watchdog (HWWD). If neither SM bit nor MM bit (bit 5 and bit 6 of CSVCR) is set, reset cause was the external reset or the hardware watchdog. If SM is '1' the reset cause is a missing Sub clock and if MM is '1' the reset cause is a missing Main clock.

16.5. Cautions

After a Clock Supervisor reset, the CLKPLL is not usable as clk source, if the clock supervisor reset was caused by a missing OSCMAIN.

Chapter 17 Clock Modulator

This chapter provides an overview of the Clock Modulator and its features. It describes the register structure and operation of the Clock Modulator.

17.1. Overview

The clock modulator is intended for the reduction of electromagnetic interference - EMI, by spreading the spectrum of the clock signal over a wide range of frequencies.

The module is fed with an unmodulated reference clock with frequency F_0 , provided by the Main PLL circuit. This reference clock is frequency modulated, controlled by a random signal.

The mean frequency of the modulated clock is equal to the reference clock frequency F_0 .

Figure 17.1-1 Block diagram of the Clock modulator

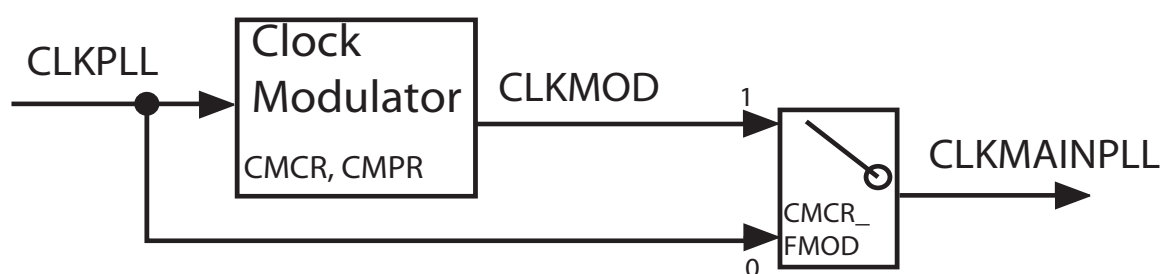
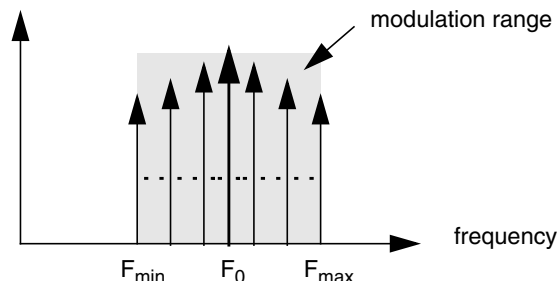


Figure 17.1-2 Frequency spectrum of the modulated clock



■ Modulation degree and frequency resolution in frequency modulation mode

Maximum and minimum frequencies (F_{max} and F_{min}) of the modulated clock are defined by the modulation degree parameter. Furthermore the resolution of the modulation range is selectable in 7 steps from low (1) to high (7). Higher resolution implies a finer granularity of discrete frequencies in the spectrum of the modulated clock but less possible modulation degrees.

In general the highest possible frequency resolution combined with the highest possible modulation degree results in the highest EMI reduction. But for some cases lower modulation degrees may result in a better EMI behavior. Refer to the table of possible settings in datasheet.

17.2. Clock Modulator Registers

This section lists the clock modulator registers and describes the function of each register in detail.

17.2.1 Clock modulator registers

Figure 17.2-1 Clock modulator registers

Address:	7	6	5	4	3	2	1	0	
0004B9 _H	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	CMPRL (lower)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Initial value 1 1 1 1 1 0 1 _B
	15	14	13	12	11	10	9	8	
0004B8 _H	-	-	MP13	MP12	MP11	MP10	MP9	MP8	CMPRH (upper)
	-	-	R/W	R/W	R/W	R/W	R/W	R/W	Initial value X X 0 0 0 0 1 0 _B
	7	6	5	4	3	2	1	0	
0004BB _H	-	Re-served	Re-served	Re-served	FMOD RUN	-	FMOD	PDX	CMCR
	-	R/W	R/W	R/W	R	-	R/W	R/W	Initial value X 0 0 1 0 X 0 0 _B

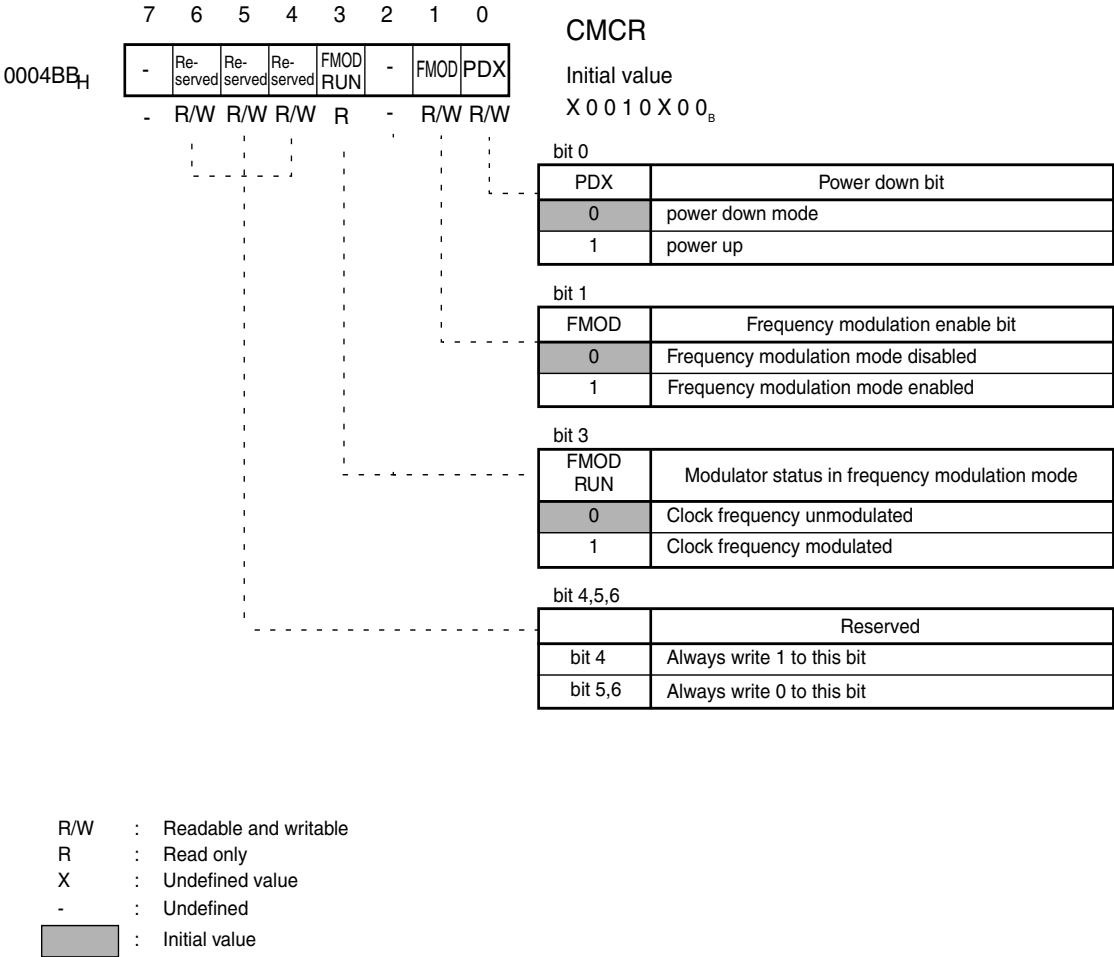
Note: For the MB91V460A Eva device there is a restriction for the input frequency of the clock modulator.
Input frequency of clock modulator is limited to the range: 16 MHz - 24 MHz.

17.2.2 Clock Modulator Control Register (CMCR)

The Control Register (CMCR) has the following functions:

- Set the modulator to power down mode
- Modulator enable/disable in frequency modulation mode
- Indicates the status of the modulator

Figure 17.2-2 Configuration of the clock modulator control register (CMCR)



The bits FMODRUN, FMOD, PDX control or indicate the status of the frequency modulation mode. Frequency modulation mode needs some additional configuration (CMPR register).

■ **Clock modulator control register contents**

Table 17.2-1 Function of each bit of the clock modulator control register (1 / 2)

Bit name		Function
bit7	undefined	
bit 6 to 5	Reserved	Always write 0 to this bit.
bit 4	Reserved	Always write 1 to this bit.
bit 3	FMOD RUN: Modulator status in frequency modulation mode bit	<p>"0": MCU is running with unmodulated clock "1": MCU is running with frequency modulated clock</p> <ul style="list-style-type: none"> • FMODRUN indicates the status of the modulator output clock in frequency modulation mode (FMOD=1). If the output clock is frequency modulated, MODRUN is set to 1, otherwise MODRUN is set to 0. • After enabling the frequency modulation mode by setting FMOD to 1, the modulator is calibrated. During this time, the clock is unmodulated. Therefore it takes several μs before the output clock switches to modulated clock and the FMODRUN bit is set to 1. The calibration time depends on the frequency of the oscillator. • During normal operation, after calibration is finished, the clock is not switched to unmodulated clock anymore. • Due to the synchronization of the FMOD signal and the synchronized switching to unmodulated clock, it takes less than $9 \times T_0$ (input clock period) before FMODRUN changes to 0 and the clock switches to unmodulated clock after the modulator is disabled. • The FMODRUN bit is read only. Writing to FMODRUN has no effect. • Before changing the parameter register CMPR, the modulator must be disabled -> FMOD=0 and FMODRUN=0.
bit 2	Undefined	

Table 17.2-1 Function of each bit of the clock modulator control register (2 / 2)

Bit name		Function
bit 1	FMOD: Frequency modulation enable bit	<p>"0": Frequency modulation disabled. "1": Frequency modulation enabled.</p> <ul style="list-style-type: none"> To enable the modulator in frequency modulation mode, FMOD must be set to 1. Before the modulator can be enabled, the PLL must deliver a stable reference clock (PLL lock time must be elapsed). Each PLL output frequency offers a set of possible modulation parameters. The selected setting (CMPR register) and the PLL frequency must match. For more information refer to the CMPR register description. Whenever the PLL output frequency is changed or the PLL is switched off e.g. in power down modes, the modulator must be disabled before -> FMOD=0 and FMODRUN=0. Before the modulator can be enabled, it must be switched from power down to active mode by setting PDX to 1. And the startup time must be awaited. Refer to the application note for a description of the recommended startup sequence. Before the modulator can be enabled in frequency modulation mode, a proper setting must be selected via the parameter register CMPR. After enabling the frequency modulation mode by setting FMOD to 1, the modulator is calibrated. During this time, the clock is unmodulated. Therefore the output clock does not switch immediately to modulated clock. The status of the clock (frequency modulated / unmodulated) is indicated by the FMODRUN status bit. Refer to the FMODRUN bit description. Due to the synchronization of the FMOD signal and the synchronized switching to unmodulated clock, it takes less than $9 \times T_0$ (input clock period) before the clock switches to unmodulated clock after the modulator is disabled. The modulator can be disabled at any time. Before changing the parameter register CMPR, the modulator must be disabled -> FMOD=0 and FMODRUN=0.
bit 0	PDX: Power down bit	<p>"0": Power down mode "1": Power up</p> <ul style="list-style-type: none"> PDX is the power down signal for the modulator. Before the frequency modulation mode can be enabled, this bit must be set to 1 and the startup time must be awaited. For more information refer to the application note for a description of the recommended startup sequence. Before switching to power down mode (PDX=0), the modulator must be disabled -> FMOD=0 and FMODRUN=0.

In the Table below the modulator states are summarized:

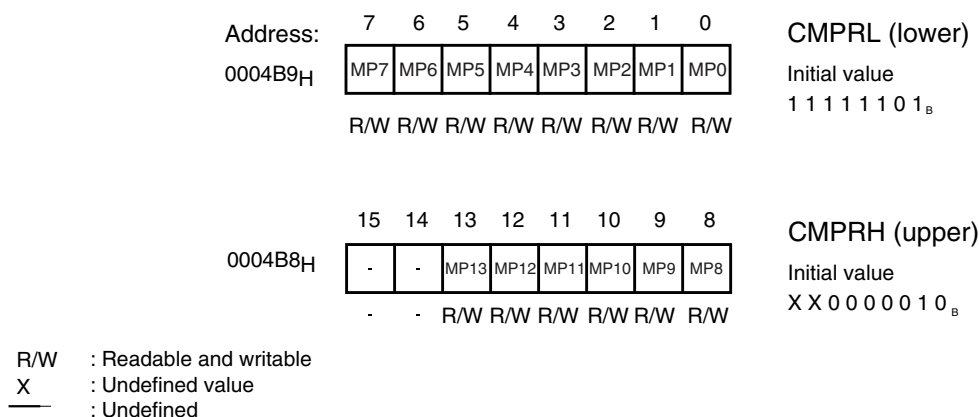
Table 17.2-2 States of the modulator

		FMOD	PDX	FMODRUN (read only)
modulator disabled		0	0	0
modulator power on, waiting modulator startup time (modulation calibration time + modulation startup time)		0	1	0
modulator enabled in frequency modulation mode, modulator is calibrating, modulation not active		1	1	0
modulator is running in frequency modulation mode modulation is active		1	1	1
	others not allowed			

17.2.3 Clock Modulation Parameter Register (CMPR)

The Modulation Parameter Register (CMPR) determines the modulation degree in frequency modulation mode.

Figure 17.2-3 Modulation parameter register



- The modulation parameter determines the degree of modulation and the maximal and minimal occurring frequencies in the modulated clock. Refer to the application note for a description of an approach to select the optimal setting.
- Each set of possible modulation parameters refers to a particular PLL frequency. The PLL frequency and the selected parameter must match. Refer to the following table of possible settings.
- The modulation parameter affects only the frequency modulation mode.

Note: The modulation parameter must be changed only when the modulator is disabled and the RUN flag is 0 (FMOD=0, FMODRUN=0).

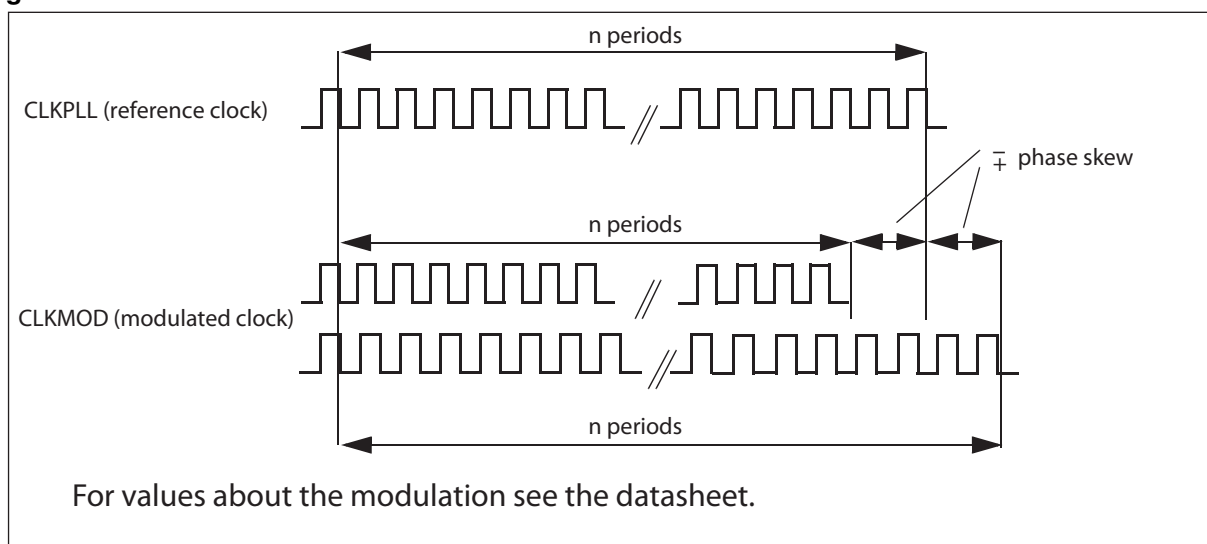
■ Modulation parameter register contents

Table 17.2-3 Function of each bit of the modulation parameter register (CMPR)

Bit name		Function
bit 15, 14	Undefined	
bit 13 to 0	MP13 to 0: Modulation Parameter bits	Depending on the PLL frequency the following modulation parameter settings are possible. The corresponding CMPR register value is stated in the most right column.

F0:	Frequency of unmodulated input clock (PLL frequency)
T0:	Period of unmodulated input clock (PLL clock period)
resolution:	resolution of frequencies in the modulated clock. low (1) to high (7)
F_{min} :	minimal frequency occurring in the frequency modulated clock
F_{max} :	maximal frequency occurring in the frequency modulated clock
phase skew:	The maximal phase shift of the modulated clock relative to the unmodulated reference clock in terms of clock periods of the unmodulated clock.
phase skew 50:	phase skew for sequences with $n \leq 50$ periods
CMPR:	register setting of the CMPR register

Figure 17.2-4 Skew of modulated clock vs. unmodulated clock



Note: **NOT ALL SETTINGS ARE ALLOWED ON EVERY DEVICE!**

Consider the actual maximal allowed clock frequency of the MCU (refer to the data sheet).
Refer to the datasheet of each device about modulation parameter settings.

17.3. Application Note

Startup/stop sequence for frequency modulation mode.

Modulation parameter for frequency modulation mode.

17.3.1 Recommended startup sequence for frequency modulation mode

- | | |
|-------|---|
| start | <ol style="list-style-type: none"> 1. Switch modulator from power down to power up mode PDX=1 2. Switch on PLL
Wait PLL for the lock time (refer to the PLLLOCK bit in 13.4.5 CSCFG: Clock Source Configuration Register (Page No.299)).
At the same time the modulator starts up. 3. Set CMPR register to a proper setting
Enable frequency modulation mode FMOD=1 4. After the calibration is finished, the clock switches from unmodulated to modulated clock and the FMODRUN flag changes to 1
... running... |
| stop | <ol style="list-style-type: none"> 5. Disable modulator FMOD=0 6. Wait until FMODRUN changes to 0 7. Switch to power down mode PDX=0 8. Disable PLL, switch to power down mode, etc. |

Note: Do not enable the modulator before the PLL lock time has elapsed. Do not disable the PLL while the modulator is running.

17.3.2 Modulation parameter for frequency modulation mode

It is not possible to recommend a particular modulation parameter setting to achieve a particular reduction in EMI. The best setting depends much on the actual application, the whole system and the requirements.

In order to find the optimal modulation parameter setting in frequency modulation mode, the following approach is recommended.

1. define the required PLL frequency based on performance needs
2. determine the maximal allowed clock frequency of the MCU
3. choose the setting with the highest resolution and the highest modulation degree, whose maximal frequency is below the maximal allowed clock frequency of the MCU.
4. perform EMI measurements
5. if the EMI measurements do not fulfill the requirements, either
 - the modulation degree at the same frequency resolution should be reduced (this may improve the reduction in the upper frequency band > 100MHz, but decrease the reduction of the fundamental < 100MHz)
 or
 - increase the modulation degree at a lower frequency resolution (this may improve the reduction of the fundamental < 100MHz, but worsen the reduction in the upper frequency band > 100MHz)
6. repeat item 3) with the new setting and continue until the best settings is identified

Chapter 18 Timebase Counter

18.1. Overview

The timebase counter is a 26-bit up-counter that counts the base clock.

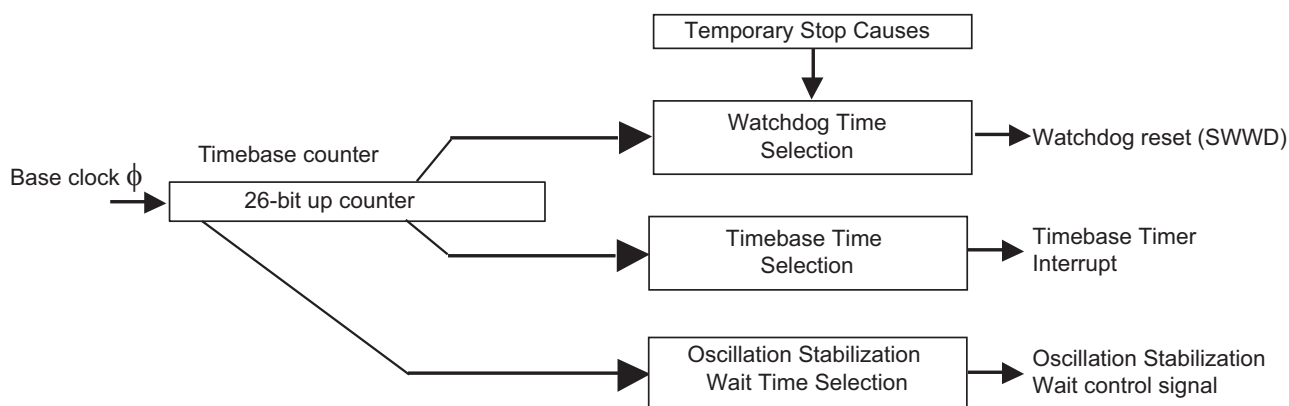
When recovering from a state in which the selected clock source for the MCU has been, or may have been, halted, the MCU automatically changes to the oscillation stabilization wait state to avoid any unstable output from the oscillator.

During the oscillation stabilization wait time, supply of internal and external clocks is halted and only the timebase counter continues to operate until the time set by the oscillation stabilization wait time setting has elapsed.

The Timebase Counter and the Timebase Timer (See "Timebase Timer" on P. 379) are based on the same hardware module. The Timebase counter is used in this chapter to generate the oscillation stabilization wait time.

Later, when the software is running the Timebase counter is used as a timer (Timebase Timer).

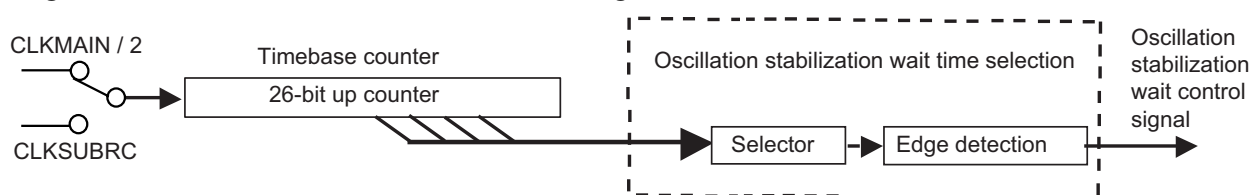
Figure 18.1-1 Timebase Counter (overview diagram)



This diagram is just an overview. Each part is explained in detail in the respective chapter.

The timebase counter, timebase timer, and watchdog timer are collectively called the watchdog control unit.

Figure 18.1-2 Timebase counter when used to generate the oscillation stabilization wait



18.2. Features

18.2.1 Timebase Counter (when used to generate the oscillation stabilization wait)

Type	: 26-bit up-counter
Clock source	: Base clock (just CLKMAIN / 2 or CLKSUBRC, because PLL is not running)
Clear	: Cleared automatically when changing to oscillation stabilization wait state.

18.2.2 Events that Invoke an Oscillation Stabilization Wait

■ Events that invoke an oscillation stabilization wait using the timebase counter

- Wait time after a settings initialization: Invoked automatically (timebase counter)
 - INITX Initial oscillation stabilization wait after pin input
 - Watchdog reset (SWWD and HWWD)
 - If the Main clock oscillation has not been halted: Oscillation stabilization wait not required
 - If the Main clock oscillation has halted: Oscillation stabilization wait is required

Example: If a watchdog reset occurs during Sub clock mode with Main clock oscillation halted

- Wait time after recovering from STOP state: Invoked automatically (timebase counter)
 - STOP state cases when clock oscillation circuit is halted:
 - The oscillation stabilization wait time for the intended oscillation circuit is required
 - Wait time for Main PLL to lock is required (if Main PLL is used)
 - STOP state cases when clock oscillation circuit is not halted:
 - Wait time for Main PLL to lock is required (if Main PLL is used)

■ Events that invoke an oscillation stabilization wait using other than the timebase counter

- Wait time after power on: Provided by INITX pin input
- Wait time after changing from Sub clock to Main clock: Using the main oscillation stabilization wait timer to generate this time is recommended.
- When recovering from Main clock oscillation halted: Enabling the Main clock oscillation and waiting for oscillation to stabilize is required.
- Main PLL lock wait time (for Main clock operation): Using the timebase timer interrupt to generate this time is recommended.
 - A wait time is required after the Main PLL operation is enabled.
 - A wait time is required after the Main PLL multiplier setting is changed.

18.3. Configuration

Figure 18.3-1 shows the configuration of the timebase counter to generate the oscillation stabilization wait time.

Figure 18.3-1 Configuration Diagram of the timebase counter for oscillation stabilization wait time

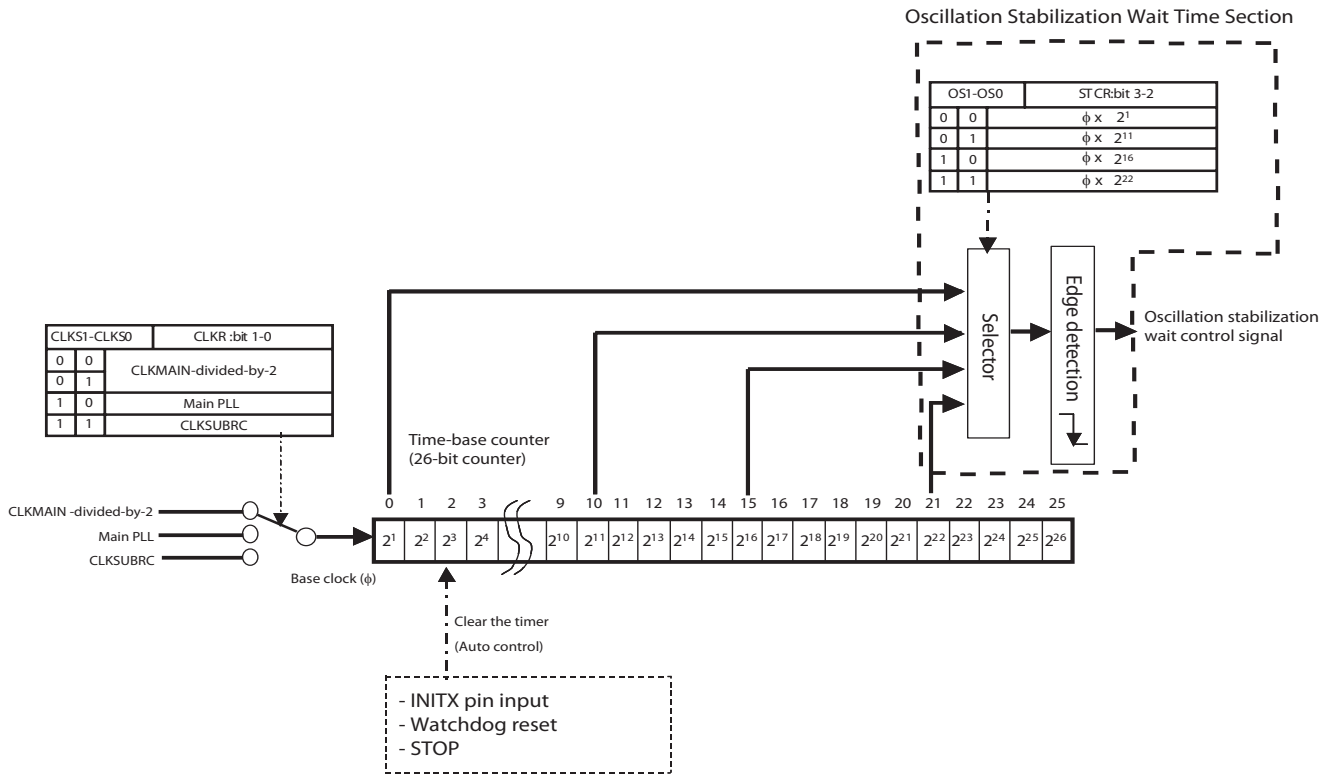


Figure 18.3-2 Register List

Time-base counter									
Address	Bit	7	6	5	4	3	2	1	0
00481H		STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1
STCR									
(Standby control)									
Clock control									
Address	Bit	7	6	5	4	3	2	1	0
00484H		---	---	---	---	SCKEN	PLL1EN	CLKS1	CLKS0
CLKR									
(Clock source control)									

18.4. Registers

18.4.1 STCR: Standby Control Register

Controls transition to standby modes, pin states during STOP state, whether to halt the clock during STOP state, the oscillation stabilization wait time, and software reset.

Note: See also “[Chapter 10 Standby \(Page No.241\)](#)” and “[Chapter 20 Software Watchdog Timer \(Page No.389\)](#)” chapters.

• **STCR: Address 0481h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1	
0	0	1	1	0	0	1	1	Initial value (INITX pin input)
0	0	1	1	X	X	1	1	Initial value (Watchdog)
0	0	X	1	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R1,W	R/W	R/W	RX,W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7: STOP state (STOP)
 - Setting “1” changes to STOP state.
- Bit6: SLEEP state (SLEEP)
 - Setting “1” changes to SLEEP state.
 - If this bit and the STOP state bit (STOP) bit are set to “1” at the same time, the device goes to STOP state.
- Bit5: High impedance mode (HIZ)
 - Setting “0” specifies that pins maintain the same states they have on entering STOP state.
 - Setting “1” specifies that pin outputs go to high impedance (Hi-z) during STOP state.
- Bit4: Software reset (SRST)
 - Setting “0” triggers a software reset.
 - Note that negative logic is used.
- Bit3-2: Oscillation stabilization time selection

OS[1:0]	The oscillation stabilization wait time after a reset (INIT) or on recovering from STOP state.		
	Oscillation stabilization wait time	When using Main clock (For a 4.0MHz Main clock)	When using Sub clock (For a 32.768kHz Sub clock)
00	$\Phi \times 2^1$	1.00μs	61μs
01	$\Phi \times 2^{11}$	1.0ms	62.5ms
10	$\Phi \times 2^{16}$	32ms	2.0s
11	$\Phi \times 2^{22}$	2s	128s

- Φ : Main clock divided by two or Sub clock
- In the case of a reset triggered by an INITX pin input, operation defaults to “00” ($\Phi \times 2^1$, Main clock).
- In the case of other resets or on recovering from STOP state, the specified clock (Main or Sub) and oscillation stabilization wait time (OS[1:0]) are used.
- The count is performed by the timebase counter.

MB91460 Series

- Bit1: Sub clock oscillation halt (OSCD2)
Setting “1” specifies that the Sub clock oscillation halts in STOP state.
- bit0: Main clock oscillation halt (OSCD1)
Setting “1” specifies that the Main clock oscillation halts in STOP state.
(See “[18.8. Caution \(Page No.378\)](#)”.)

18.4.2 CLKR: Clock Source Control Register

Selects the clock source for the base clock used to run the MCU and controls the PLL.

Note: See also the “[Chapter 13 Clock Control \(Page No.289\)](#)”.

• **CLKR: Address 0484h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	-	-	SCKEN	PLL1EN	CLKS1	CLKS0	
X	X	X	X	0	0	0	0	Initial value (INITX pin input)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-4: Reserved bit Always write “0” to this bit. The read value is the value written.
- Bit3: Sub clock selection enable (SCKEN)
 - Setting this bit to “1” enables the Sub clock to be selected.
- Bit2: Main PLL operation enable (PLL1EN)
 - Setting this bit to “1” starts Main PLL operation. Main PLL can be selected as the operating clock after the Main PLL has locked.
- Bit1-0: Clock source selection

CLKS1	CLKS0	Clock source setting	Mode
0	0	The Main clock input from X0/X1 divided by 2 (initial value)	Main clock mode
0	1	The Main clock input from X0/X1 divided by 2	Main clock mode
1	0	Main PLL	Main clock mode
1	1	Sub clock	Sub clock mode

When changing the clock mode, the value of CLKS0 cannot be modified if CLKS1 is “1”

The table below lists the cases when the CLKS1 - CLKS0 bits may or may not be modified.

Table 18.4-1 Cases When the CLKS1 and CLKS0 Bits May or May Not be Modified

Modify permitted	Modify not permitted
“00” -> “01” or “10”	“00” -> “11”
“01” -> “11” or “00”	“01” -> “10”
“10” -> “00”	“10” -> “01” or “11”
“11” -> “01”	“11” -> “00” or “10”

Example: To select the Sub clock after an INIT reset, first write “01_B” and then write “11_B” (Sub clock)

The clock source for the timebase counter during the oscillation stabilization wait time is set by the clock source selection bits.

CLKS1	CLKS0	Clock source for timebase counter during oscillation stabilization wait time	Mode
0	0	The Main clock input from X0/X1 divided by 2 (initial value)	Main clock mode
0	1		
1	0 ^{*1}		
1	1	Sub clock	Sub clock mode

1. The PLL is not active during oscillation stabilization time, therefore the setting ‘01’ also selects Main clock divided by 2.

18.5. Operation

This section describes the events that trigger an oscillation stabilization wait and the operation in each case.

18.5.1 INITX Pin Input

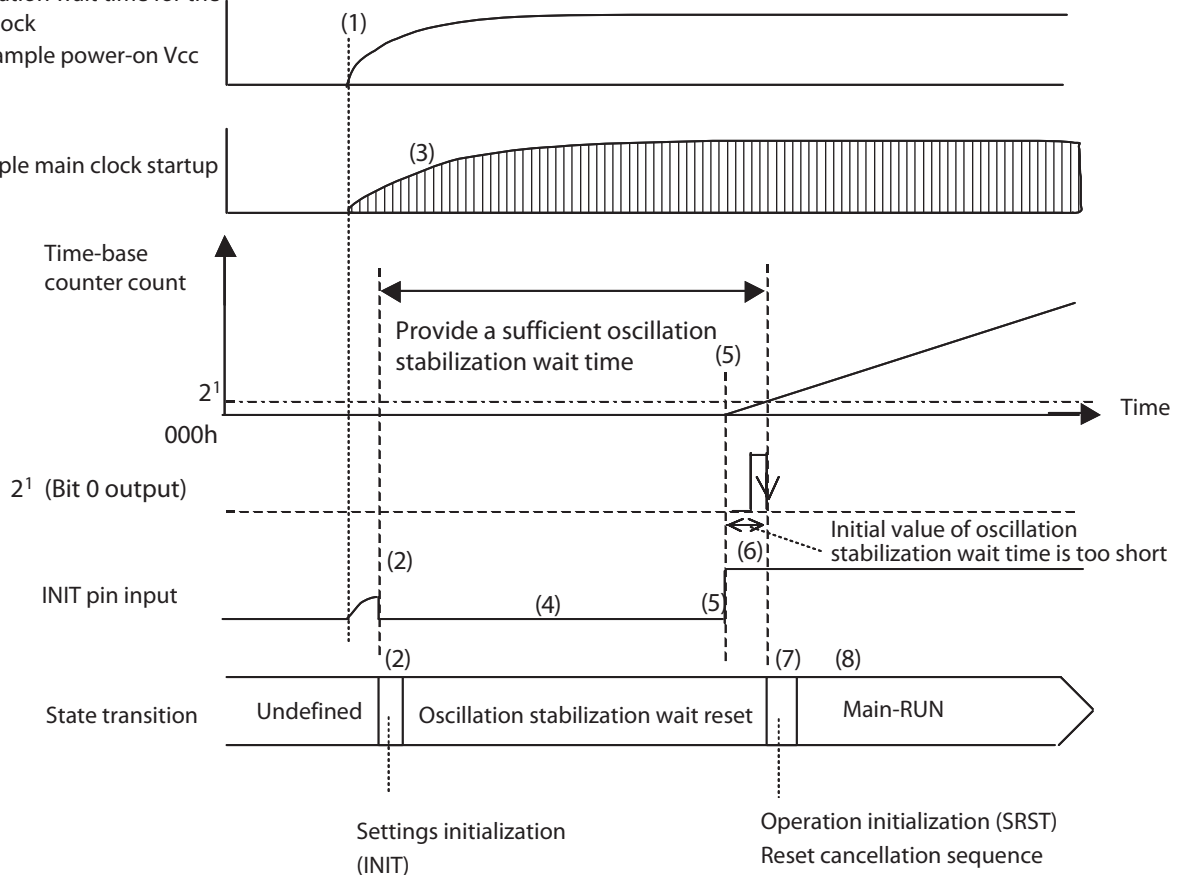
An oscillation stabilization wait is required after power on. As the wait time provided by the initialized timebase counter is too short, the INITX pin input must be held at the “L” level.

Figure 18.5-1 Using the Width of the Pin Input to Provide the Oscillation Stabilization Wait Time

Using the INIT pin input to trigger a reset and provide the oscillation stabilization wait time for the main clock

Example power-on Vcc

Example main clock startup



- (1) Power turned on
- (2) Start INITX pin input (Settings initialization reset)
- (3) Main clock oscillation starting
- (4) INITX pin input (to provide a sufficient time for the Main clock oscillation to stabilize)
- (5) INITX pin input removed. The timebase counter is initialized and starts counting.
- (6) Oscillation stabilization wait time provided by timebase timer/counter (Initial value = minimum value)
(If the INITX pin input (4) is not maintained, the wait time is too short.)
- (7) Operation initialization reset, reset cancellation sequence
- (8) Main-RUN

■ INITX Pin input when Main clock running

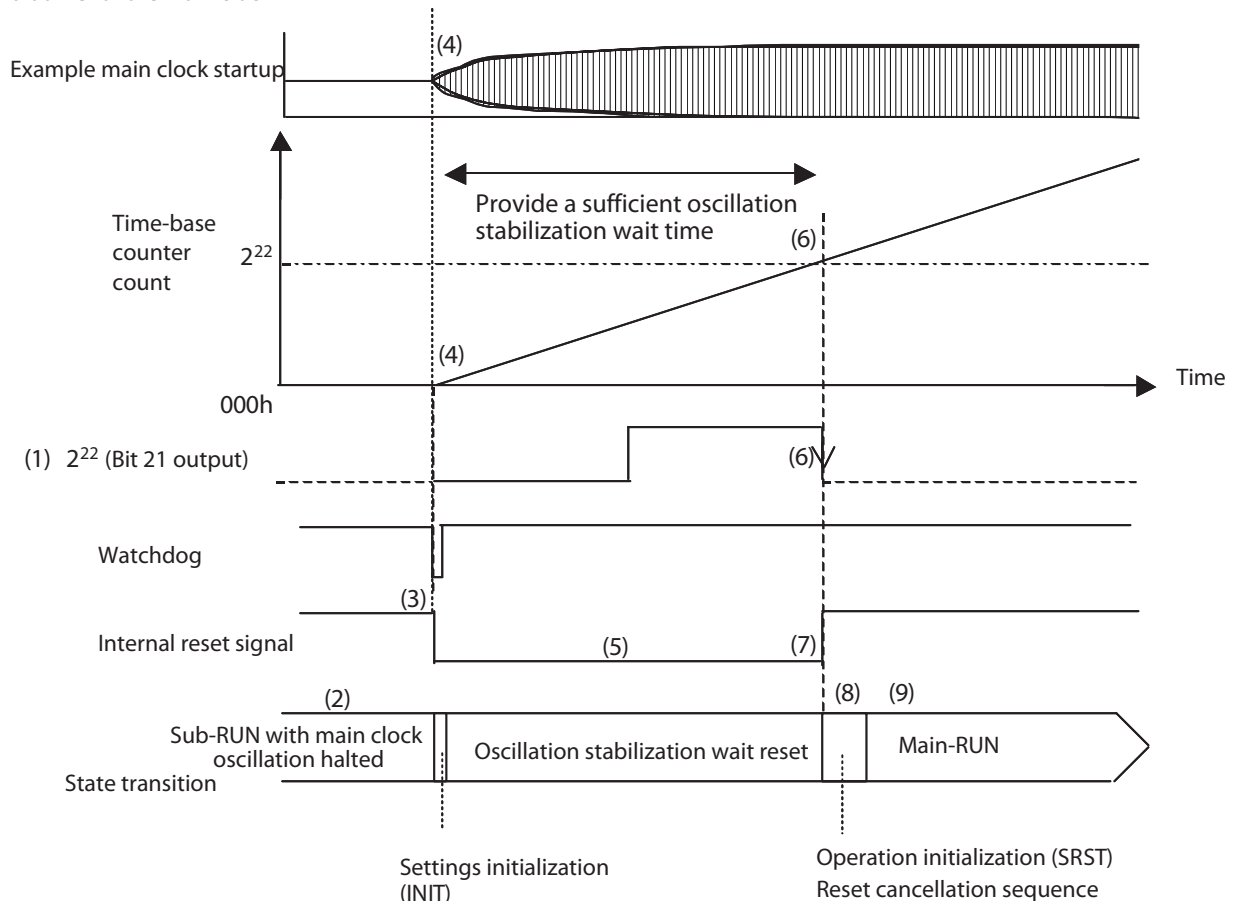
The device goes to the operation initialization reset (RST) state automatically after the minimum oscillation stabilization wait time elapses.

18.5.2 Watchdog Reset - HWWD and SWWD (The specified oscillation stabilization wait time is generated automatically)

If a watchdog reset occurs while the Main clock oscillation is halted, the oscillation stabilization wait time is generated automatically. (See figure below.)

Figure 18.5-2 Watchdog Reset when Main Clock Halted (Sub-RUN)

Using the time-base counter to provide the oscillation stabilization wait time for the main clock



- (1) Oscillation stabilization wait time selection (Example: Main clock divided by two x 2^{22})
(Set the interval time beforehand to provide an adequate oscillation stabilization wait time.)
- (2) Sub-RUN with Main clock oscillation halted
- (3) Watchdog reset occurs
- (4) Main clock oscillation starts
The timebase counter is cleared and starts counting.
- (5) Oscillation stabilization wait
- (6) Time set as the timebase timer interval time (time set in (1))
- (7) Reset released, operation initialization (SRST)
- (8) Operation initialization, reset sequence
- (9) Main-RUN

Note: If a watchdog reset occurs when the Main clock oscillation is halted during Sub clock mode (Sub clock is being used as clock source) by the Main clock oscillation halt bit (OSCCR.OSCDS1), the device changes to the oscillation stabilization wait state after the settings initialization reset (INIT) is released. The device

then changes to the operation initialization reset (RST) state after the oscillation stabilization wait time elapses.

■ Watchdog reset when Main clock operating

Although no oscillation stabilization wait is required in this case, the specified wait time is generated automatically.

18.5.3 Recovering from STOP state via an Interrupt

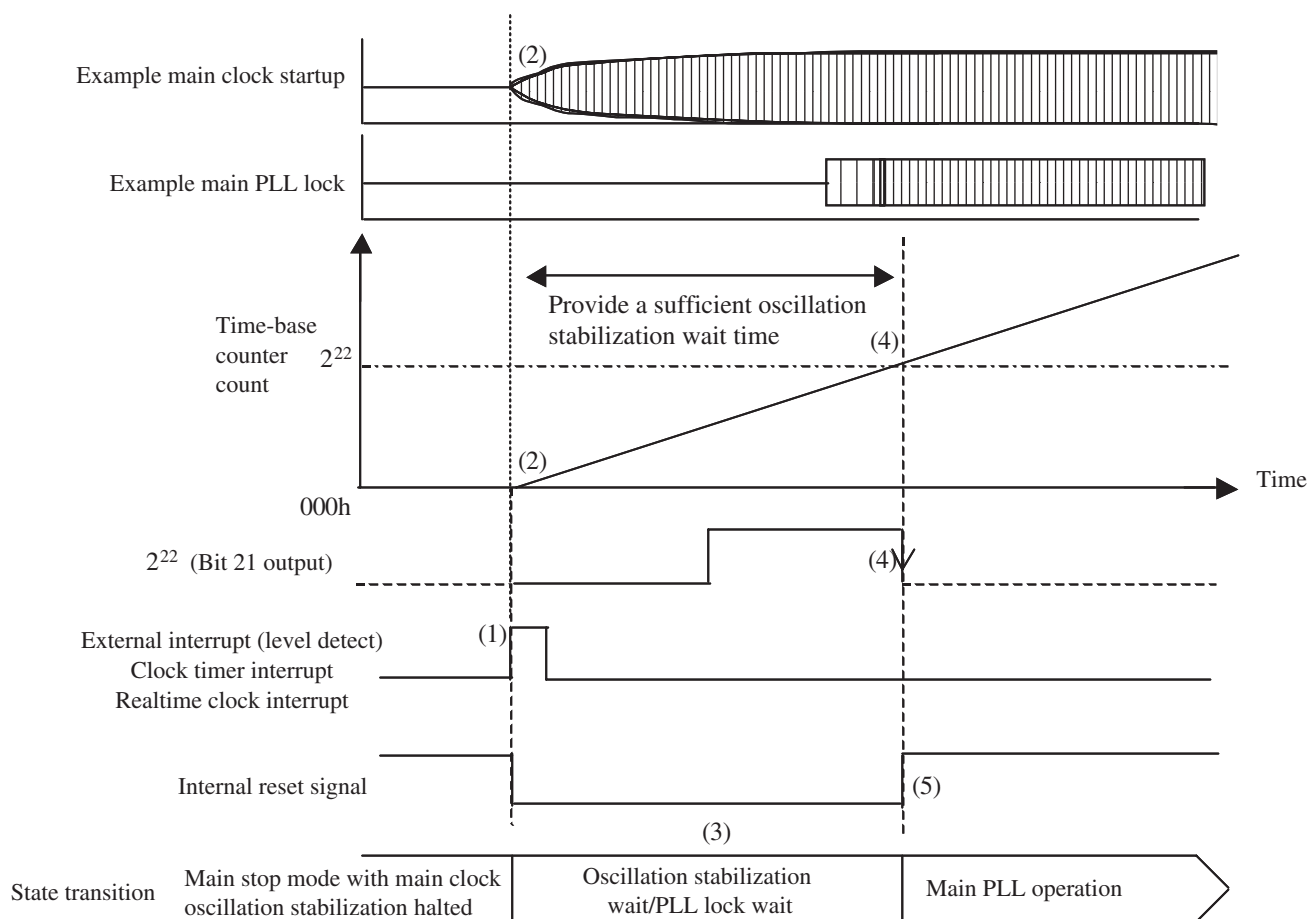
■ When changing from Main PLL operation to STOP state with the Main clock oscillation halted (STCR.OSCD[2:1]="11"):

The Main oscillation circuit generates the selected oscillation stabilization time automatically.

Figure 18.5-3 shows the recovering from STOP state with the Main Clock halted to main PLL operation via an Interrupt

Figure 18.5-3 Recovering from STOP state to Main PLL Operation via an Interrupt

Usint the time-base counter to generate oscillation stabilization wait
Main clock/Main PLL



■ When changing to STOP state without halting the clock oscillation circuit (Main PLL/Main/Sub):

Although no oscillation stabilization wait is required in this case, a wait is generated automatically. Accordingly, it is recommended that the interval time is set to its minimum value before changing to STOP state.

- When recovering from STOP state, the device goes to the oscillation stabilization wait state immediately after STOP state is released.
- The next state after the oscillation stabilization wait completes depends on what triggered recovery from STOP state.
- If recovery was triggered by an enabled external interrupt, sub oscillation stabilization timer interrupt, or main oscillation stabilization wait timer interrupt, the device goes to the normal operating state (RUN).

Note: If the Main PLL continues to operate in STOP state, changing to STOP state with the Main PLL clock set as the active clock is not permitted. Always set the active clock to the Main clock divided by two or the Sub clock beforehand.)

18.5.4 The lock wait time for the Main PLL must be generated by software.

■ Wait time after Main PLL operation enabled:

Using the timebase timer interrupt is recommended

However, the Main PLL must not be selected as the clock source.

■ Wait time after Main PLL multiplier modified:

Using the timebase timer interrupt is recommended

However, the Main PLL must not be selected as the clock source.

See “[Chapter 19 Timebase Timer \(Page No.379\)](#)” for details.

18.5.5 Generating an Oscillation Stabilization Wait when Changing from Sub clock mode to Main Clock Mode

■ When Main clock continues to run during Sub clock mode:

- If not using Main PLL after changing clock: No oscillation stabilization wait time
- If using Main PLL after changing clock: Main PLL lock wait is required.
(Using the timebase timer interrupt is recommended. See “[18.5.3 Recovering from STOP state via an Interrupt \(Page No.371\)](#)”.)

■ When Main clock halts during Sub clock mode:

- A Main clock oscillation stabilization wait is required before changing clock.
(Use the oscillation stabilization wait timer for the Main clock. See “[Chapter 22 Main Oscillation Stabilization Timer \(Page No.405\)](#)”.)
- When using the Main PLL: A further wait is required for the Main PLL to lock.
(Using the timebase timer interrupt is recommended. See “[18.5.3 Recovering from STOP state via an Interrupt \(Page No.371\)](#)”.)

18.5.6 When Recovering from an Abnormal State with the Main PLL Selected

When the Main PLL is set as the clock source and a problem of some sort occurs in Main PLL control (such as the multiplier setting being changed or the Main PLL enable bit modified during Main PLL operation), the device goes to the oscillation stabilization wait state automatically to provide the Main PLL lock time. The device then goes to normal operating mode after the oscillation stabilization wait elapses.

MB91460 Series

18.5.7 Types of Oscillation Stabilization Wait

■ Timebase counter

Automatically provides a count for the oscillation stabilization wait time.

When a trigger occurs to change the device to the oscillation stabilization wait state, the timebase counter is cleared and then starts counting the specified oscillation stabilization wait time.

■ “L” level input to INITX pin

When device operation is restarted by inputting an “L” level to the INITX pin while the oscillation is halted (the three cases listed below), the width of the “L” level input provides the stabilization time required by the oscillation circuit.

- INITX pin input after power turned on.
- INITX pin input when oscillation halted during STOP state
- INITX pin input when Sub clock selected as clock source and Main clock oscillation halted.

■ Timebase timer

Using the timebase timer to generate the Main PLL lock time is recommended.

See “[Chapter 19 Timebase Timer \(Page No.379\)](#)” for details.

■ Main oscillation stabilization wait timer

Used when restarting the Main clock while operating in Sub clock mode.

See “[Chapter 22 Main Oscillation Stabilization Timer \(Page No.405\)](#)” for details.

■ Sub oscillation stabilization wait timer

Used when restarting the Sub clock while operating in Main clock mode.

See “[Chapter 23 Sub Oscillation Stabilization Timer \(Page No.417\)](#)” for details.

18.5.8 Whether or not a Stabilization Wait is Required for Each State Transition

Figure 18.5-4 Stabilization Wait for each state transition (necessary or not)

Reset due to INIT pin input

State	PLL	Main	Sub	Operation after INIT signal input	Is oscillation stabilization wait required ?	Remarks
Wait time after power on	X	X	X	Operate using initial value (CLKMAIN/2)	Must be provided by width of INIT pin input	The automatic oscillation stabilization wait (minimum value) is too short
Main clock running (= main clock oscillation running)	X	0	0	Operate using initial value (CLKMAIN/2)	Oscillation stabilization wait not required	Uses the automatic oscillation stabilization wait (initial value = minimum value)
Main PLL running (= main PLL oscillation running)	0	0	0			
Main stop (main clock/main PLL oscillation running)	0/X	0	0			
Sub clock running (main clock/main PLL oscillation running)	0/X	0	0			
Sub stop (main clock oscillation running)	0/X	0	0			
From main clock (1/2) running to main stop (main clock halted)	X	X	0	Operate using initial value (CLKMAIN/2)	Must be provided by width of INIT pin input	The automatic oscillation stabilization wait (minimum value) is too short
From main PLL running to main stop (Main clock halted)	X	X	0			
Sub clock running (main clock halted)	X	X	0			
Sub stop / sub sleep (main clock halted)	X	X	0			
Main clock oscillation stabilization wait in progress	X	X	0/X			

Watchdog reset

State	PLL	Main	Sub	Operation after INIT signal input	Is oscillation stabilization wait required ?	Remarks
Main clock running (= main clock oscillation running)	X	0	0	Operate using initial value (CLKMAIN/2)	Oscillation stabilization wait not required	Uses the automatic oscillation stabilization wait. (The oscillation stabilization wait time is not initialized)
Main PLL running (= main PLL oscillation running)	0	0	0			
Sub clock running (main clock/main PLL oscillation running)	0/X	0	0			
Sub clock running (main clock halted)	X	X	0	Operate using initial value (CLKMAIN/2)	Main clock oscillation stabilization wait is required	Uses the automatic oscillation stabilization wait. (Set an appropriate wait time setting)

Recovery from STOP state via an Interrupt

State	PLL	Main	Sub	Op. State after recovering from STOP state via Interrupt	Is oscillation stabilization wait required ?	Remarks
Main-STOP (CLKMAIN continues during STOP state, main PLL running) ¹⁾	0	0	0	Previous operation state (CLKMAIN/2)	When changing to main PLL after recovering, a main PLL lock wait time is required	Uses the automatic oscillation stabilization wait. Set wait time setting to OS[1:0] = "11" to provide the main PLL lock time.
Main-STOP (CLKMAIN continues during STOP state, main PLL halted)	X	0	0	Previous operation state (CLKMAIN/2)	Oscillation stabilization wait not required	Uses the automatic oscillation stabilization wait. Set wait time setting to minimum value.
Main-STOP (CLKMAIN halted during STOP state (automatic))	X	X	0	Previous operation state (CLKMAIN/2)	Main clock oscillation and main PLL lock wait required	Uses the automatic oscillation stabilization wait. Set an appropriate wait time setting.
Sub-STOP (CLKMAIN and main PLL running)	0/X	0	0	Previous operation state (Sub)	Oscillation stabilization wait not required	Uses the automatic oscillation stabilization wait. Set wait time setting to minimum value. (Before changing to CLKMAIN an oscillation stabilization wait is required)
Sub-STOP (CLKMAIN halted)	X	X	0			

¹⁾ The active clock must be set to CLKMAIN/2 before changing to STOP state.

Main clock oscillation enable

State	PLL	Main	Sub	Main clock oscillation enabled	Is oscillation stabilization wait required ?	Remarks
CLKSUB running (CLKMAIN halted)	X	X	0	Start Main clock oscillation	Main clock oscillation stabilization wait is required.	It is recommended to use the main clock oscillation stabilization wait time to generate the time.

Main PLL oscillation enable

State	PLL	Main	Sub	Main PLL oscillation enabled	Is oscillation stabilization wait required ?	Remarks
CLKSUB running (CLKMAN running)	X	0	0	Start Main clock oscillation/Change PLL multiplier setting	Main PLL lock wait required	It is recommended to use the time-base timer to generate the Main PLL lock wait time.
CLKMAIN/2 running	X	0	0			

0: Oscillation running, X: Oscillation halted

MB91460 Series**18.6. Settings****Table 18.6-1 Settings Required to Specify the Oscillation Stabilization Wait Time**

Setting	Setting register	Setting procedure*
Oscillation stabilization wait time setting	Standby control register (STCR)	See 18.4.1

*: For the setting procedure, refer to the section indicated by the number.

Table 18.6-2 Settings Required to Setup an INITX Pin Reset

Setting	Setting item	Setting procedure
INITX pin input	Refer to the oscillator parameters and the reset parameters in the Data Sheet.	—

- Settings required to specify the oscillation stabilization wait time for the Main clock
See “[Chapter 22 Main Oscillation Stabilization Timer \(Page No.405\)](#)”.
- Settings required to specify the PLL lock wait time
See “[Chapter 19 Timebase Timer \(Page No.379\)](#)”.

18.7. Q&A

18.7.1 How to setup the oscillation stabilization wait time that is generated automatically

Use the oscillation stabilization wait time selection bits (STCR.OS[1:0]). (The following lists likely scenarios and the required settings.)

Scenario	Oscillation stabilization wait time selection bits (OS[1:0])	Example oscillation stabilization wait time after a reset (INIT) or on recovering from STOP state		
		Oscillation stabilization wait time	4.0MHz Main clock running	32.768kHz Sub clock running
To not halt the Main PLL or oscillator during STOP state (No oscillation stabilization wait time required)	Set "00".	$\Phi \times 2^1$	1.00 μ s	61 μ s
To not stop the oscillator during external clock input or STOP state (Main PLL lock wait time)	Set "01".	$\Phi \times 2^{11}$	1.0ms	62.5ms
When using an oscillator with a fast stabilization time such as a ceramic resonator (Oscillation stabilization wait time (medium))	Set to "10".	$\Phi \times 2^{16}$	32ms	2.0s
When using a standard quartz oscillator (Oscillation stabilization wait time (long))	Set to "11".	$\Phi \times 2^{22}$	2s	128s

- Φ : Main clock divided by 2, or Sub clock
- In the case of an INITX pin input, operation defaults to "00" ($\Phi \times 2^1$ = Main clock divided by 4).
- For other resets and when recovering from STOP state, the operation is in accordance with the specified clock (Main or Sub) and oscillation stabilization wait time (OS[1:0]) setting.
- The count is performed by the timebase counter.
- Once the time is selected, it is not initialized except by a settings initialization triggered by the external INITX pin.

MB91460 Series**18.7.2 How to set the oscillation stabilization wait time without generating it automatically**

The settings described below for various cases are required.

State (before transition)	Oscillation			Condition (after transition)	Is oscillation stabilization wait required?	To set the oscillation stabilization wait time
	PLL	Main	Sub			
Wait time after power on	×	×	×	Operation after INITX pin input defaults to Main clock (1/2) (Initial value)	Main clock oscillation stabilization wait is required	As the automatic oscillation stabilization wait (minimum value) is too short, the width of the INITX pin must be sufficient to provide the stabilization time.
Sub clock operation (Main clock halted)	×	×	O			
Sub-SLEEP, Sub-STOP (Main clock halted)	×	×	O			
Main clock oscillation stabilization wait	×	×	O/×			

State (before transition)	Oscillation			Main clock oscillation enabled	Is oscillation stabilization wait required?	To set the oscillation stabilization wait time
	PLL	Main	Sub			
Sub clock operation (Main clock halted)	×	×	O	Main clock running	Main clock oscillation stabilization wait is required	Using the Main clock oscillation stabilization wait timer to generate the time is recommended.

State (before transition)	Oscillation			Main PLL running and enabled	Is oscillation stabilization wait required?	To set the oscillation stabilization wait time
	PLL	Main	Sub			
Sub clock running (Main clock running)	×	O	O	Start Main PLL oscillation/ Change PLL multiplier setting	Main PLL lock wait required	Using the timebase timer to generate the Main PLL lock wait time is recommended.
Main clock (1/2) running	×	O	O			

O: Oscillation running, ×: Oscillation halted

18.7.3 What is the clear timing for the timebase counter?

- The timebase counter is only cleared automatically by INITX pin input, and by watchdog reset (SWWD and HWWD)

The timebase counter automatically starts counting after being cleared.

- The timebase counter can also be cleared by software.

See “[Chapter 19 Timebase Timer \(Page No.379\)](#)” for details.

18.8. Caution

- Clock source

If the clock selected as the clock source is not stable, an oscillation stabilization wait time is required.

- Oscillation stabilization wait time

The wait time set in the oscillation stabilization time selection bits (STCR.OS[1:0]) is not initialized by any reset except a reset triggered by the external INITX pin input, the RC based watchdog or the Clock Supervisor. For other resets including settings initialization resets (timebase counter based watchdog reset) and operation initialization resets (RST), the wait time set prior to the reset is used.

- Watchdog reset (Timebase Counter based watchdog)

Although an oscillation stabilization wait time is not required if a watchdog reset occurs while the Main clock is running (Main or Sub), a wait time is generated automatically. In this case, the oscillation stabilization wait time (STCR.OS[1:0]) is not initialized.

- “L” level input to INITX pin

As the oscillation stabilization wait time is initialized to its minimum value when an initialization is triggered by an INITX pin input, the wait time in this case is too short. Ensure the INITX pin input width is long enough to provide the oscillation stabilization wait time.

In the following three cases, maintain the INITX pin input at the “L” level for long enough to provide the oscillation stabilization wait time required by the oscillation circuit.

- INITX pin input after turning on the power
- INITX pin input after oscillation halted in STOP state
- INITX pin input when Sub clock selected as the clock source and Main clock oscillation halted

(Accordingly, to stabilize the oscillation of both the Main and Sub clocks, input an “L” level to the INITX pin for long enough to provide a sufficient oscillation stabilization time for both the Main and Sub clocks.)

- Main PLL lock wait

If enabling the Main PLL from the halted state after program execution starts, the Main PLL must not be used until after sufficient time has elapsed for the Main PLL to lock.

Similarly, when changing the multiplier setting for the Main PLL when the PLL is running, the new Main PLL clock must not be used until sufficient time has elapsed for the Main PLL to lock.

Using the timebase timer interrupt to generate the Main PLL lock wait time is recommended.

- Cases when oscillation stabilization wait is not required

Although no oscillation stabilization wait is required when recovering via an interrupt from Main-STOP or Sub-STOP state when the Main clock oscillation has not been halted, the oscillation stabilization wait is generated automatically. Setting the wait time to its minimum value prior to entering STOP state is recommended.

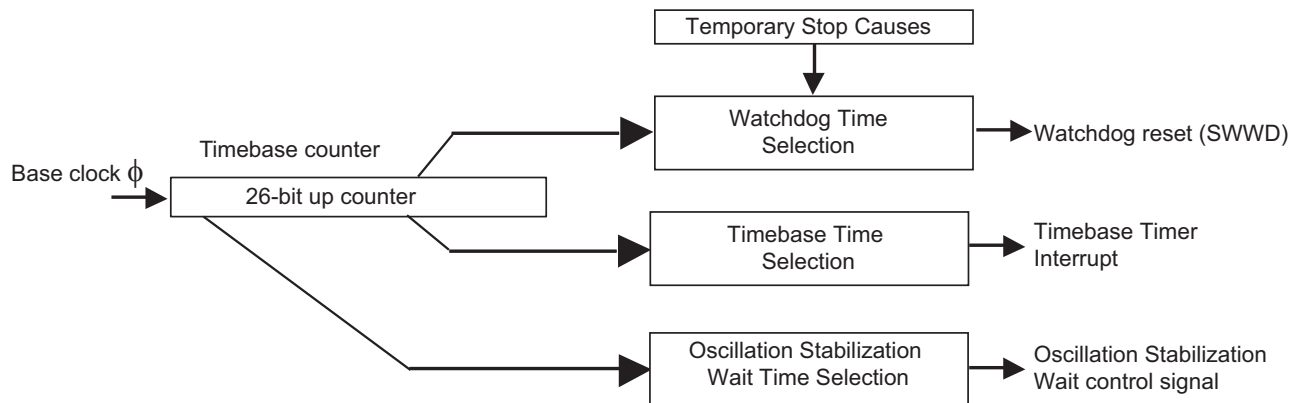
Chapter 19 Timebase Timer

19.1. Overview

The Timebase Timer and the Timebase Counter (Chapter 18 Timebase Counter (Page No.363)) are based on the same hardware module. The Timebase counter is used in this chapter as an Timebase timer.

The timebase timer is an interval-interrupt generating timer that is used to acquire Main PLL lock wait time and to count a long time.

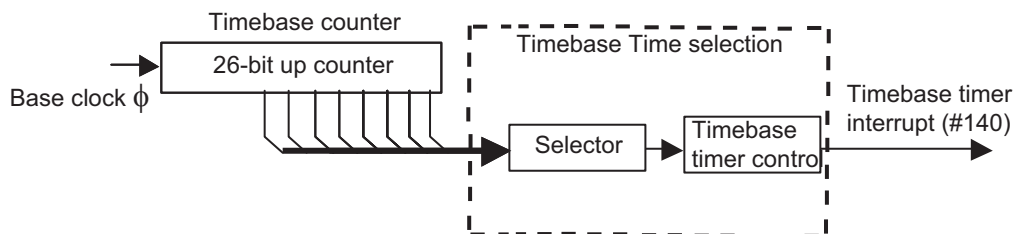
Figure 19.1-1 Timebase Counter (overview diagram)



This diagram is just an overview. Each part is explained in detail in the respective chapter.

The timebase counter, timebase timer, and watchdog timer are collectively called the watchdog control unit.

Figure 19.1-2 Timebase Counter used to generate the Timebase Timer Interrupt



19.2. Features

■ Timebase timer (TBT)

- Type : Detects timebase timer bit output and generates an interval interrupt.
- Interval time : 8 types (Timebase timer bit output)

$$\text{Period} = \phi \times 2^{11}, \phi \times 2^{12}, \phi \times 2^{13}, \phi \times 2^{22}, \phi \times 2^{23}, \phi \times 2^{24}, \phi \times 2^{25}, \phi \times 2^{26}$$
- Operation start/stop: Always in operation (Can be replaced by interrupt request enable control)
- Timebase counter clear: Continuously writes "A5" "5A" in the timebase counter clear register CTBR using the software.

19.3. Configuration

Figure 19.3-1 Configuration Diagram of the Timebase Timer

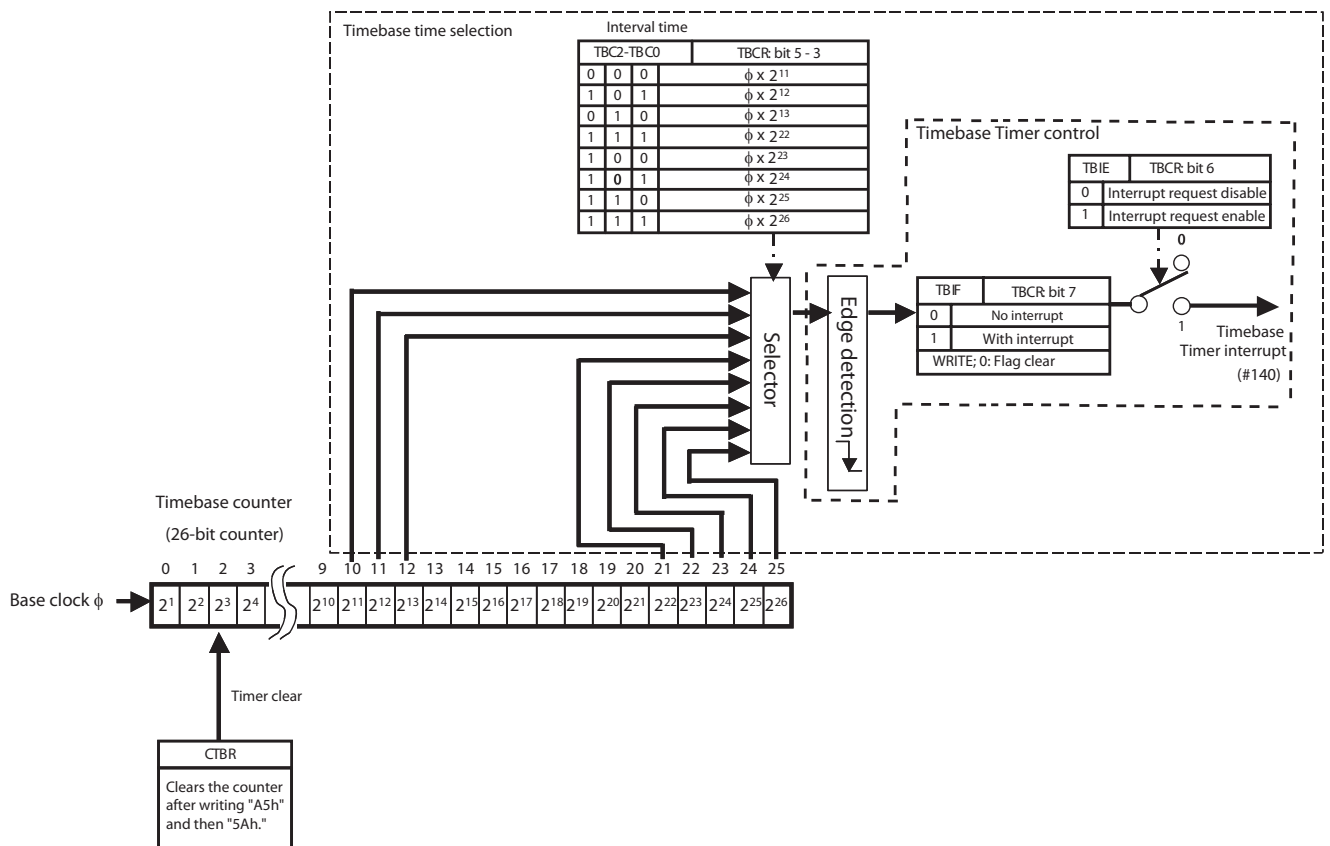


Figure 19.3-2 List of Registers

Timebase timer

Address	Bit	7	6	5	4	3	2	1	0		
00482H		TBIF	TBIE	TBC2	TBC1	TBC0	---	SYNCR	SYNCS	TBCR	Timebase counter control
00483H		D7	D6	D5	D4	D3	D2	D1	D0	CTBR	Timebase counter clear
0045FH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR62	Interrupt level
0FFF40H		32Bits									Interrupt vector #140

MB91460 Series

19.4. Register

19.4.1 TBCR: Timebase Timer Control Register

This register is used to set timebase timer interrupt control, reset/ standby operation option etc.

Note: Refer also to “[Chapter 10 Standby \(Page No.241\)](#)”.

- **TBCR: Address 0482h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
TBIF	TBIE	TBC2	TBC1	TBC0	---	SYNCR	SYNCS	
0	0	X	X	X	X	0	0	Initial value (INITX pin input, watchdog reset)
0	0	X	X	X	X	X	X	Initial value (the software reset)
R(RM1), W	R/W	R/W	R/W	R/W	RX/WX	RX/WX	R/W	Attribute

(Refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for the attributes.)

- Bit7: Timebase timer interrupt flag

TBIF	Operation	
	Read	Write
0	With no interrupt	Flag is cleared
1	With interrupt (The interval time set by the timebase timer has elapsed)	Writing does not affect operation

- An interrupt request is generated if the timebase timer interrupt request enable bit is “1”, and if the timebase timer interrupt flag is “1”.

- Bit6: Timebase timer interrupt request enable

TBIE	Operation
0	Disabling the timebase timer interrupt request
1	Enabling the timebase timer interrupt request

- Bit5-3: Selecting the timebase timer interval time

TBC2-TBC0	Interval time	Example	
		(32.0MHz, base clock)	(32.768kHz, base clock)
000	$\Phi \times 2^{11}$	64.0μs	62.5ms
001	$\Phi \times 2^{12}$	128μs	125ms
010	$\Phi \times 2^{13}$	256μs	250ms
011	$\Phi \times 2^{22}$	131ms	128s
100	$\Phi \times 2^{23}$	262ms	256s
101	$\Phi \times 2^{24}$	524ms	512s
110	$\Phi \times 2^{25}$	1048ms	1024s
111	$\Phi \times 2^{26}$	2097ms	2048s

- Be sure to set the interval time before an interrupt.
(Oscillation stability wait time used when returning to STOP caused by an interrupt)
- Bit2: Reserved bit. Writing does not affect the operation. The read value is indefinite.

- Bit1: Enabling the synchronous reset operation

SYNCR	Operation
0	Asynchronous reset operation
1	Synchronous reset operation enable

- Ordinary operation reset: Immediately resets the operation initialization when the operation initialization reset (RST) request is generated.
Synchronous reset: Resets the operation initialization after all accesses to the bus have stopped.
- Bit0: Synchronous standby operation enable

SYNCS	Operation
0	Asynchronous reset operation (Not permitted on this model).
1	Enable synchronous standby operation (always set this before changing to a standby mode).

19.4.2 CTBR: Timebase Counter Clear Register

This register is used to initialize the timebase counter.

- **CTBR: Address 0483h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	Attribute

(Refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for the attributes)

- Continuously writing “A5_H”, “5A_H” in the timebase counter clear register clears the timebase counter immediately after writing “5A_H”. (All bits are “0”)
There is no time restrictions between “A5_H” and “5A_H”, but if “A5_H” is written followed by the one other than “5A_H”, it must be written “A5_H” again. If not, the timebase counter cannot be cleared even if “5A_H” is written.
- The read value is indefinite.
- Clearing the timebase counter using the timebase counter clear register temporarily modifies the relevant items shown below.
 - Oscillation stability wait interval
 - Watchdog timer period
 - Timebase timer period

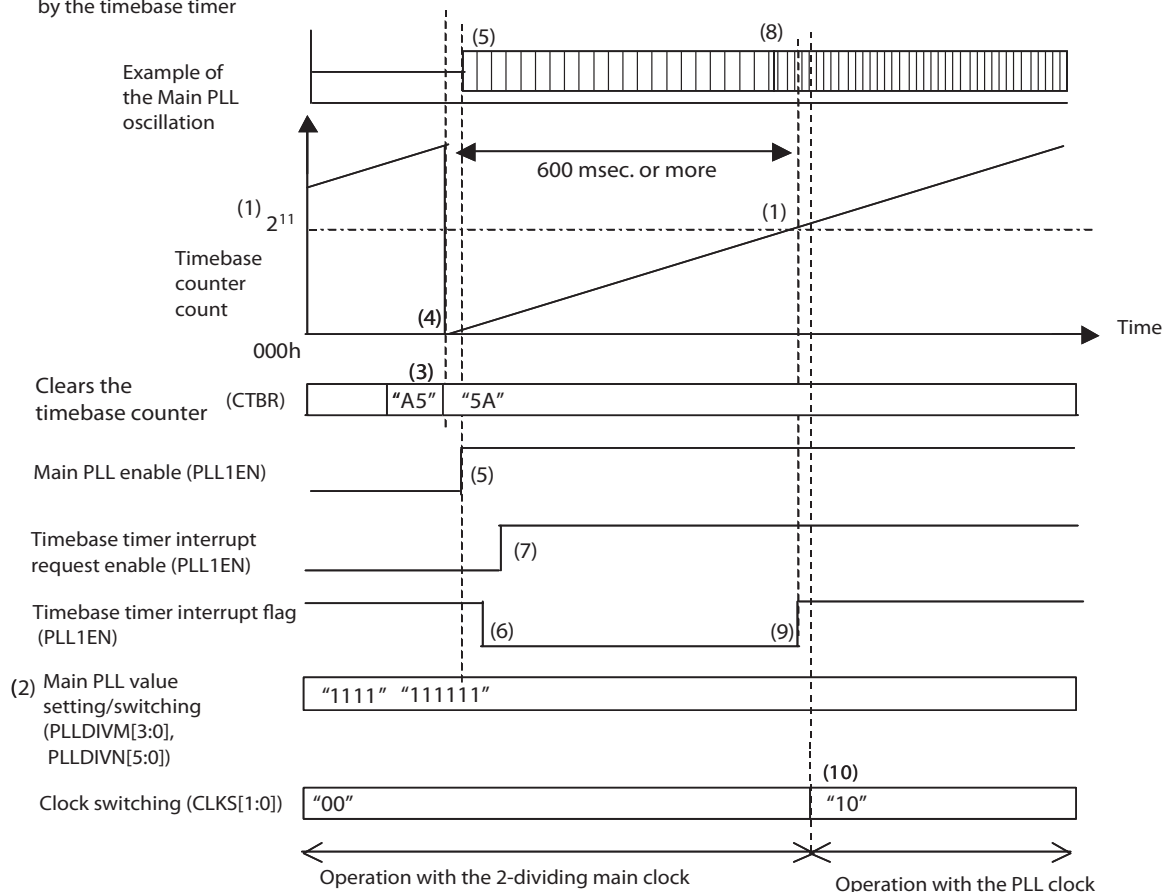
19.5. Operation

Timebase timer operation is described.

19.5.1 Timebase Timer Interrupt Example (Main PLL Lock Wait)

Figure 19.5-1 Example for Timebase Timer Interrupt (Main PLL Lock Wait)

Main PLL lock wait
by the timebase timer



- (1) Selecting the interval value in the timebase timer
- (2) Selecting the Main PLL value (Setting / Switching)
- (3) Writing data in the timebase counter clear register in the order of "A5" and "5A"
- (4) Writing "5A" clears the timebase counter and causes the count to start from "0"
- (5) Enables the Main PLL to operate
- (6) Clears the timebase timer interrupt flag using the software
- (7) Setting the timebase timer interrupt request enable bit to "1"
- (8) The Main PLL locks
- (9) A timebase timer interrupt occurs when the timebase timer interval time has elapsed
- (10) Setting the Main PLL to the operation clock

19.6. Setting

Table 19.6-1 Setting Required for the Timebase Timer

Setting	Setting register	Setting method*
Setting the interval time	Timebase timer control register control register (TBCR)	Refer to 19.4.1
Timebase counter clear	Timebase counter clear register (CTBR)	Refer to 19.4.2

*: Refer to the number for more information on the setting method.

Table 19.6-2 Setting Required for Interrupting the Timebase Timer

Setting	Setting register	Setting method*
Setting the timebase timer interrupt vector and interrupt level	Refer to “ Chapter 24 Interrupt Control (Page No.429) ”	Refer to 24.4.3
Setting the Main clock oscillation stability wait timer interrupt Interrupt request clear Interrupt request enable	Timebase timer control register control register (TBCR)	Refer to 19.4.1

*: Refer to the number for more information on the setting method.

MB91460 Series

19.7. Q & A

19.7.1 What are the types of interval time used in the timebase timer (and the timebase counter used by the timebase timer) and how to select them?

There are eight types of interval time, and they are set using the interval selection bit (TBCR.TBC[2:0]).

Timebase timer Interval time	Interval selection bit (TBC[2:0])	Example Interval time		
		$\Phi = 2\text{MHz}$	$\Phi = 32\text{MHz}$	$\Phi = 32.768\text{kHz}$
How to select $\Phi \times 2^{11}$	Set the value to "000"	1.024ms	64 μs	62.5ms
How to select $\Phi \times 2^{12}$	Set the value to "001"	2.048ms	128 μs	125ms
How to select $\Phi \times 2^{13}$	Set the value to "010"	4.096ms	256 μs	256ms
How to select $\Phi \times 2^{22}$	Set the value to "011"	2.097s	131ms	128s
How to select $\Phi \times 2^{23}$	Set the value to "100"	4.194s	262ms	256s
How to select $\Phi \times 2^{24}$	Set the value to "101"	8.388s	524ms	512s
How to select $\Phi \times 2^{25}$	Set the value to "110"	16.77s	1.04s	1024s
How to select $\Phi \times 2^{26}$	Set the value to "111"	33.55s	2.09s	2048s

ϕ : This is the base clock. (Refer to "Chapter 13 Clock Control (Page No.289)".)

19.7.2 What Is the count clock of the timebase counter?

The count clock is a base clock. Refer to "Chapter 13 Clock Control (Page No.289)".

19.7.3 How to operate the timebase timer

The timebase timer is always operating. (Setting is unnecessary.)

However, to use interval interrupt, interrupt setting is required.

19.7.4 How is the timebase timer (=timebase counter) operation stopped?

It cannot be stopped.

19.7.5 How is the timebase counter (=timebase timer) cleared?

If {A5_H} and {5A_H} is written successively in the timebase counter clear register CTBR, the timebase counter is cleared immediately after {5A_H}. (All bits are "0".)

However, if the timebase counter is cleared, the watchdog timer is affected. (Refer to "19.8. Caution (Page No.387)")

19.7.6 How about the interrupt-associated registers?

Setting timebase timer's interrupt vector and interrupt level

The relationship between the interrupt level and vector is shown in the following table.

Refer to "Chapter 24 Interrupt Control (Page No.429)" for more information on the interrupt level and interrupt vector.

Interrupt vector (default)	Interrupt level setting bit (ICR[4:0])
#140 Address: 0FFDCCh	Interrupt level register (ICR62) Address: 047Eh

The interrupt flag (TBCR.TBIF) cannot automatically be cleared. As a result, clear it by the software before returning from an interrupt service. (Write "0" in the interrupt flag (TBIF).)

19.7.7 What are the interrupt types?

One type of interrupt is available, and an interrupt is generated when the interval time that is set using the interval selection bit (TBCR.TBC[2:0]) has elapsed. (Selection is unnecessary.)

19.7.8 How is an interrupt enabled?

Interrupt request enable bit and interrupt flag

To enable the interrupt request the interrupt request enable bit is used (TBCR.TBIE).

	Interrupt request enable bit (TBIE)
Interrupt disable	Set the value to "0"
Interrupt enable	Set the value to "1"

Clearing the interrupt request is performed using the interrupt flag (TBCR.TBIF).

	Interrupt flag (TBIF)
Interrupt flag clear	"0" is written

19.8. Caution

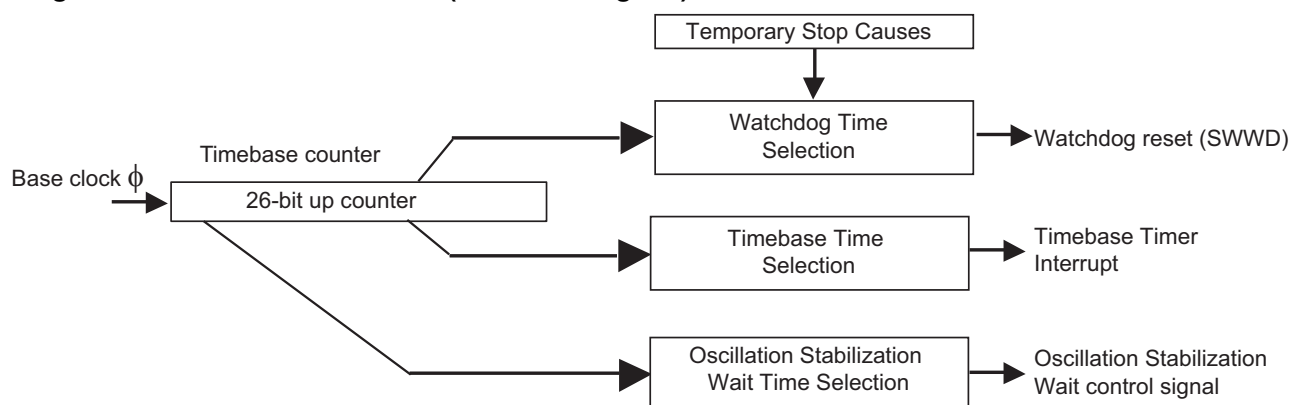
- The Main PLL needs the PLL lock wait time after operation enable and after modifying the rate of multiply. We recommend that this Main PLL lock wait time be acquired using the timebase interrupt. Make sure that the PLL lock wait time is set. (For detailed information refer to the datasheet)
- Regarding the interval setting
 - When modifying the timebase timer interval time, set the interrupt request enable bit (TBIE) to “0” in advance to disable an interrupt.
 - The timebase counter is always counting. Clear the timebase counter before enabling an interrupt to acquire an accurate interval interrupt time using the timebase timer.
(If not, an interrupt request may be generated immediately after an interrupt request enable.)
- About clearing the timebase counter using a program
 - If there is data written in the timebase counter clear register CTBR in the order of “A5_H” and “5A_H” the timebase counter is cleared immediately after writing “5A_H”. (All bits are “0.”)
 - Although there are no restrictions on the write timings for “A5_H” and “5A_H” writing “A5_H” followed by a one other than “5A_H” the clearing operation is not performed if “A5_H” is not written again even if “5A_H” is written.
 - If the timebase counter is cleared, the reset signal to the watchdog is generated with a delay once.
- About clearing the timebase counter by the hardware
The timebase counter is cleared by the STOP state and the setting initialization reset (INITX pin input, watchdog reset). (All bits are “0.”)
- In the STOP state
When returning to the interrupt from a STOP, the timebase counter is used to acquire the clock oscillation stability wait time. As a result, there is a possibility to unintentionally generate timebase timer's interval interrupt. Therefore, disable the timebase timer interrupt not to use the timebase timer before the STOP is set.

Chapter 20 Software Watchdog Timer

20.1. Overview

The software watchdog timer consists of a selector that uses the output from the 26-bit Timebase counter (Chapter 18 Timebase Counter (Page No.363)) and a one-bit counter.

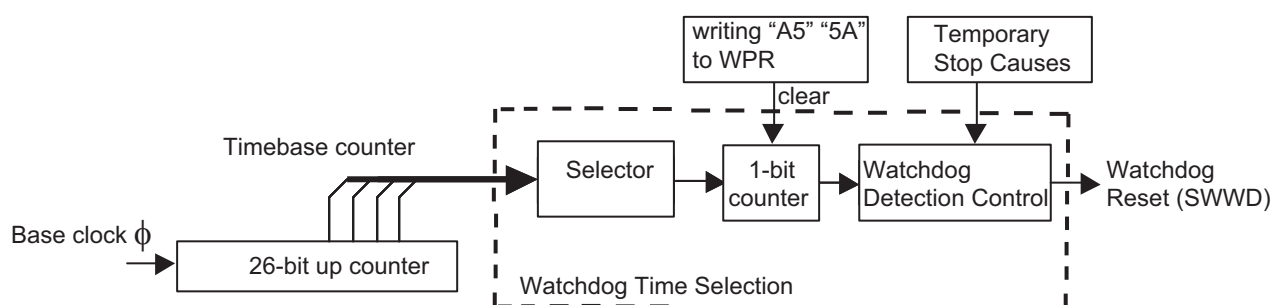
Figure 20.1-1 Timebase Counter (overview diagram)



This diagram is just an overview. Each part is explained in detail in the respective chapter.

The timebase counter, timebase timer, and watchdog timer are collectively called the watchdog control unit.

Figure 20.1-2 Timebase Counter used to generate the Watchdog Reset



20.2. Features

■ Watchdog timer

- Type : Generates the watchdog reset (INIT) with the overflow from one-bit counter
(See "RSRR: Reset Cause Register" on P. 229)
- Count clock (interval time): Bit output from the timebase timer
4 types $\phi \times 2^{20}$, $\phi \times 2^{22}$, $\phi \times 2^{24}$, $\phi \times 2^{26}$
(Can be set only once after the reset (RST).)
- Clearing 1-bit counter:
Successively writes "A5" "5A" to watchdog reset generation delay register WPR by the software. This operation does not influence the Timebase counter.
- Operation start/stop: This timer starts to operate once it writes data to the watchdog control register RSRR for the first time after the reset (RST). This timer stops by reset (RST, INIT, SWWD reset, HWWD reset).

20.3. Configuration

Figure 20.3-1 Configuration Diagram of the Watchdog Timer

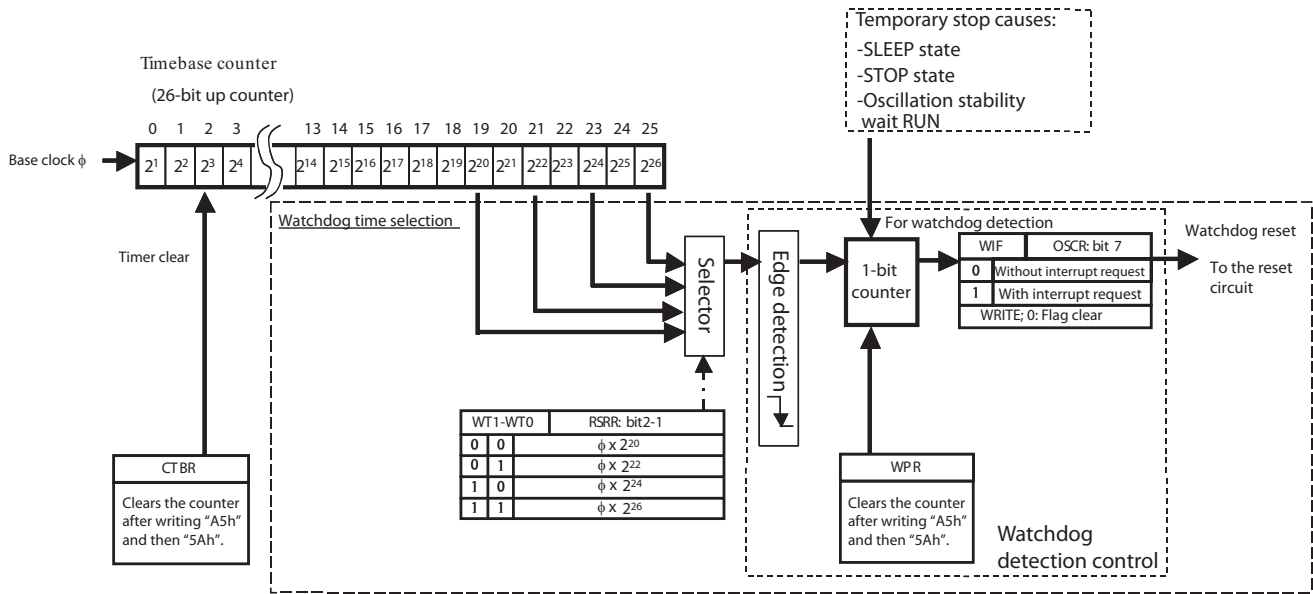


Figure 20.3-2 List of Registers

Watchdog timer

Address	Bit	7	6	5	4	3	2	1	0		
00480H		INIT	HSTB	WDOG	ERST	SRST	LINIT	WT1	WT0	RSRR	Watchdog timer control
00485H		D7	D6	D5	D4	D3	D2	D1	D0	WPR	Watchdog reset generation delay
00483H		D7	D6	D5	D4	D3	D2	D1	D0	CTBR	Time-base counter clear

MB91460 Series

20.4. Register

20.4.1 RSRR: Watchdog Timer Control Register

This register is used to set watchdog timer periods, and execute the startup control.

(This register also functions as the reset cause register that stores previously generated reset causes.)

Note: Refer also to “[Chapter 9 Reset \(Page No.227\)](#)”.

- **RSRR: Address 0480h (Access: Byte, Half-word)**

7	6	5	4	3	2	1	0	bit
INIT	HSTB	WDOG	ERST	SRST	LINIT	WT1	WT0	
1	0	0	0	0	0	0	0	Initial value (INITX pin input)
-	-	-	X	X	-	0	0	Initial value (Watchdog reset)
X	X	X	-	-	X	0	0	Initial value (Software reset)
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	Attribute

(For the attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

The watchdog timer starts once it writes the watchdog timer control register.

- **Bit7: Initialization reset occurred flag**

Indicates whether a reset (INIT) was triggered by INITX pin input.

INIT	Meaning
0	No INIT has been triggered by the INITX pin input.
1	INIT has been triggered by the INITX pin input.

The initialization reset occurred flag (INIT) is cleared to “0” after reading.

- **Bit6: Hardware Standby reset occurred flag**

Indicates whether a reset (INIT) was triggered by HSTX pin input.

HSTB	Meaning
0	No INIT has been triggered by the HSTX pin input.
1	INIT has been triggered by the HSTX pin input.

The hardware standby reset occurred flag (HSTB) is cleared to “0” after reading.

- **Bit5: Watchdog reset occurred flag**

Indicates whether a reset (INIT) was triggered by the watchdog timer.

WDOG	Meaning
0	No INIT has been triggered by the watchdog timer.
1	INIT has been triggered by the watchdog timer.

The watchdog reset occurred flag (WDOG) is cleared to “0” after reading.

- **Bit4: External reset occurred flag**

Indicates whether a reset (RST) was triggered by the RSTX pin input.

ERST	Meaning
0	No RST has been triggered by the RSTX pin input.
1	RST has been triggered by the RSTX pin input.

The external reset occurred flag (ERST) is cleared to “0” after reading.

- **Bit3: Software reset occurred flag**

Indicates whether a software reset has been triggered by writing to the software reset bit (STCR.SRST).

SRST	Meaning
0	No RST has been triggered by a software reset.
1	RST has been triggered by a software reset.

The software reset occurred flag (SRST) is cleared to "0" after reading.

- Bit2: Low voltage reset occurred flag

Indicates whether a reset (INIT) was triggered by the low voltage detection.

LINIT	Meaning
0	No INIT has been triggered by the low voltage detection.
1	INIT has been triggered by the low voltage detection.

The low voltage reset occurred flag (LINIT) is cleared to "0" after reading.

- Bit1-0: Watchdog interval time selection

WT1	WT0	The minimum writing interval required for WPR so that the watchdog timer may not be reset	Interval between the time when WPR is last written with 5A _H and when the watchdog is reset
		(Interval time of the timebase counter selection bit)	(Watchdog interval time)
0	0	$\Phi \times 2^{20}$ (Initial value)	$\Phi \times 2^{20}$ to $\Phi \times 2^{21}$
0	1	$\Phi \times 2^{22}$	$\Phi \times 2^{22}$ to $\Phi \times 2^{23}$
1	0	$\Phi \times 2^{24}$	$\Phi \times 2^{24}$ to $\Phi \times 2^{25}$
1	1	$\Phi \times 2^{26}$	$\Phi \times 2^{26}$ to $\Phi \times 2^{27}$

(Φ : Base clock)

- A total of four watchdog interval times are available to be selected.
- Only the data firstly written after a reset is valid, and the other data sets are invalid.
- Watchdog interval time selection bit can be read to know the set value.

Note: For more information on bits used for timers other than the watchdog timer, refer to "[Chapter 9 Reset \(Page No.227\)](#)".

20.4.2 WPR: Watchdog Reset Generation Postponement Register

This register is used to postpone the generation of watchdog reset.

- **WPR: Address 0485h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value (INIT)
X	X	X	X	X	X	X	X	Initial value (RST)
RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	RX,W	Attribute

(Refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for the attributes.)

- If “A5_H” and “5A_H” are successively written in the watchdog reset generation postponement register and immediately after writing “5A_H” the 1-bit counter used to detect the watchdog is set to “0” to postpone the generation of a watchdog reset.
Although there are no restrictions on the write timings for “A5_H” and “5A_H”, if “A5_H” and a value other than “5A_H” are written, “A5_H” must be written again. If not, writing “5A_H” does not set the 1-bit counter to “0”.
- The read value is indefinite.
- Both “A5_H” and “5A_H” must be written within the specified interval as shown below to prevent the watchdog reset from being generated. The intervals are shown in the following table according to the watchdog interval time selection bit (RSRR.WT[1:0]).

WT1	WT0	Minimum interval required for writing data in WPR
0	0	Within $\Phi \times 2^{20}$ (Initial value)
0	1	Within $\Phi \times 2^{22}$
1	0	Within $\Phi \times 2^{24}$
1	1	Within $\Phi \times 2^{26}$

20.4.3 CTBR: Timebase Counter Clear Register

This register is used to initialize the timebase counter.

- **CTBR: Address 0483h (Access: Byte)**

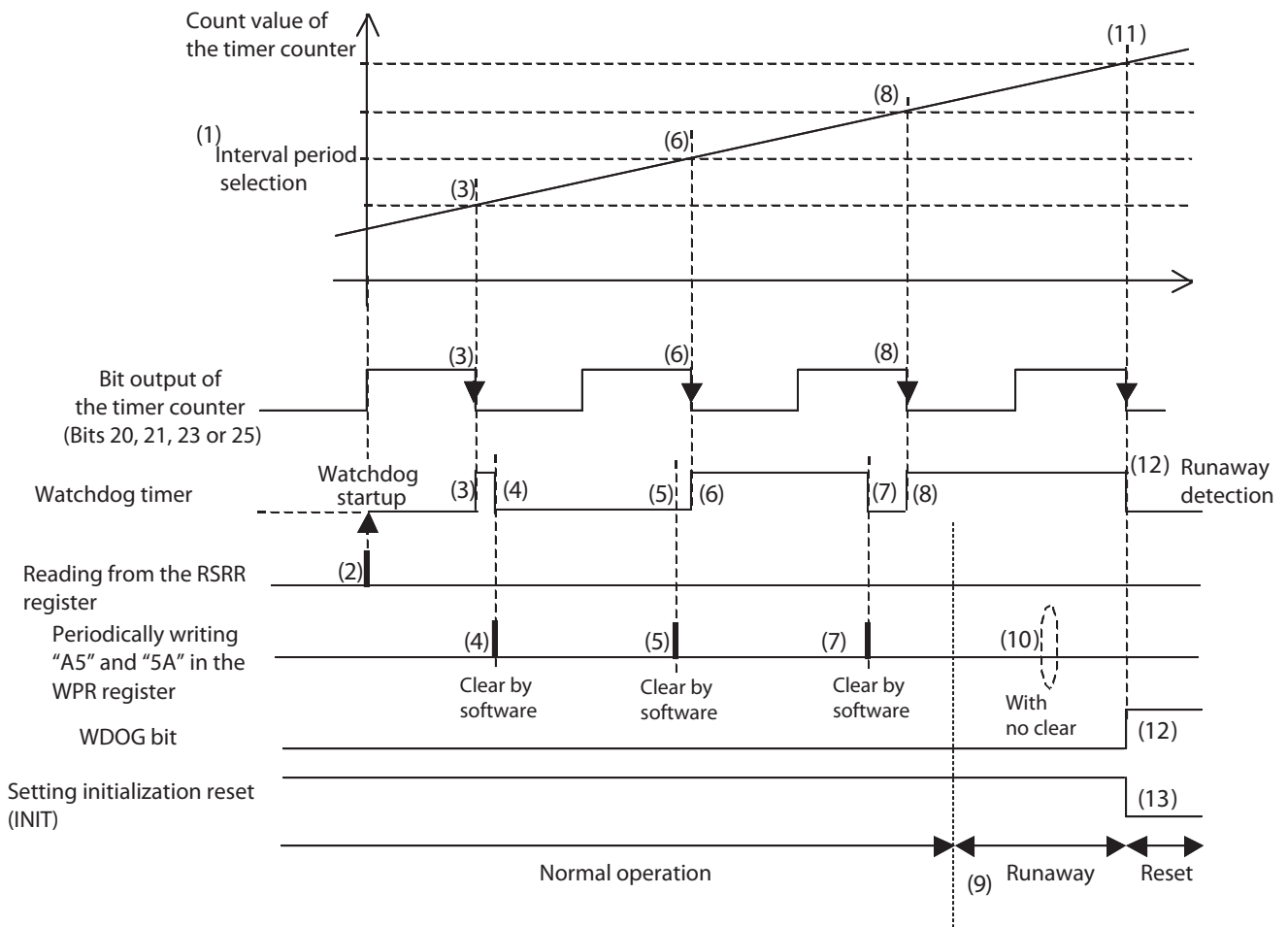
For more information, refer to “[Chapter 19 Timebase Timer \(Page No.379\)](#)”.

20.5. Operation

This section describes the watchdog operation.

20.5.1 Watchdog (Detecting Runaway)

Figure 20.5-1 Example of Watchdog operation (Detecting Runaway)



- (1) Setting interval time
- (2) Watchdog startup (Watchdog timer clear)
- (3) Interval signal output from the timebase counter. The watchdog timer counts.
- (4) Within the interval time, by software periodic writing to the WPR register with "A5" and "5A" has been performed. The watchdog timer clears.
- (5) Within the interval time, by software periodic writing to the WPR register with "A5" and "5A" has been performed. The watchdog timer clears.
- (6) Interval signal output from the timebase counter. The watchdog timer counts.
- (7) Within the interval time, by software periodic writing to the WPR register with "A5" and "5A" has been performed. Watchdog timer clears.
- (8) Interval signal output from the timebase counter. The watchdog timer counts.
- (9) MCU runs away (the runaway of MCU is assumed).
- (10) Within interval time, by software writing to the WPR register with "A5" and "5A" has not been performed.
- (11) Interval signal output from the timebase counter. The watchdog timer counts.
- (12) Runaway is detected, WODOG flag has been changed to "1".
- (13) Watchdog reset (INIT) has been generated.

20.5.2 Starting the Watchdog Timer and Setting the Watchdog Timer Period

The watchdog timer starts once it first writes data to the RSRR (Reset cause register/Watchdog timer control register) after the reset (RST). At this time, Bits 1 and 0 (WT1 and WT0 bits) set the watchdog timer interval time. Only the setting for the interval time executed first after the reset is valid, and the other settings executed at a later time are invalid.

20.5.3 Postponing the Generation of a Watchdog Reset

Once watchdog timer is started, it is necessary that the WPR (watchdog reset generation postponement register) should be written periodically with {A5_H} and {5A_H} in this order by software. This operation is used to set the 1-bit counter for detecting the watchdog reset to "0".

On MB91V460A and MB91461, the watchdog timer will be retriggered / cleared automatically by DMA during D-bus access.

20.5.4 Generation of the watchdog reset

The 1-bit counter for detecting the watchdog reset is set at the falling edge of the output of the timebase counter where an interval is set. In addition, if the second falling edge is detected while the 1-bit counter is set, the request for the setting initialization reset (INIT) is generated as the watchdog reset.

20.5.5 Temporarily Stopped Watchdog Timer (Automatic Generation Postponement)

The watchdog timer resets the 1-bit counter used for detecting the watchdog reset to "0" as initialization while CPU program operation is stopped. In this state, the generation of the watchdog reset is postponed. The states where programs stop running are concretely shown below.

- SLEEP state
- STOP state
- Oscillation stability wait RUN
- Is in break when using the emulator debugger and monitor debugger, only if DSU4 is mounted
- Is in break when using the embedded debug support unit (only if EDSU and EMMODE is enabled)
- Period between the time when executing the INTE command and when executing RETI, only if DSU4 is mounted
- Step trace trap (break per each command with PS register T flag="1"), only if DSU4 is mounted

In addition, clearing the timebase counter simultaneously initializes the 1-bit counter used for detecting the watchdog reset, thus causing the reset timing of the watchdog to be postponed.

20.5.6 Stopping the Watchdog Timer

Once the watchdog timer is started, the watchdog timer operation cannot be stopped until the initialization reset (RST) is generated.

The watchdog timer is stopped under these states shown below where the operation initialization reset (RST) is generated until it is restarted by software.

- Operation initialization reset (RST)
- Setting initialization reset (INIT)
- Oscillation stabilization wait reset

20.5.7 Re-triggering the Watchdog Timer (MB91V460A, MB91461)

On MB91V460A and MB91461, the watchdog timer will be retriggered / cleared automatically by DMA during D-bus access.

20.6. Setting

Table 20.6-1 Setting Required for Using the Watchdog Timer

Setting	Setting register	Setting method*
Interval time setting	Watchdog timer control register (RSRR)	Refer to 20.4.1
Startup of the watchdog		Refer to 20.4.1

*: Refer to the number for more information on the setting method.

Table 20.6-2 Setting Required for Delaying the Generation of the Watchdog

Setting	Setting register	Setting method*
Setting required for delay the generation of the watchdog reset	Watchdog reset generation delay register (WPR)	Refer to 20.4.2

*: Refer to the number for more information on the setting method.

Table 20.6-3 Setting Required for Checking the Generation of the Watchdog

Setting	Setting register	Setting method*
Watchdog generation check	Watchdog timer control register (RSRR)	Refer to 20.4.1

*: Refer to the number for more information on the setting method.

20.7. Q & A

20.7.1 What are the types of watchdog interval time and how are they selected?

There are four types of the interval period, and they are set using the interval selection bit (RSRR.WT[1:0]).

Watchdog Interval time	Interval Selection bit (WT[1:0])	Example) Interval Time		
		$\Phi = 80.0\text{MHz}$	$\Phi = 2.00\text{MHz}$	$\Phi = 32.768\text{kHz}$
To select $\Phi \times 2^{20}$	Set the value to "00"	13.1 ms	0.524 s	32.0 s
To select $\Phi \times 2^{22}$	Set the value to "01"	52.4 ms	2.097 s	128.0 s
To select $\Phi \times 2^{24}$	Set the value to "10"	209.7 ms	8.388 s	512.0 s
To select $\Phi \times 2^{26}$	Set the value to "11"	838.8 ms	33.554 s	2048.0 s

Note: ϕ : Base clock. (Refer to "Chapter 13 Clock Control (Page No.289)".)

Only the data sets first written after the reset (INITX pin input, watchdog reset, software reset) are valid, and the other data sets are invalid.

20.7.2 How is the watchdog operation started (set to valid)?

Writing data in the watchdog timer control register RSRR causes the watchdog timer to be started (set RSRR.WT to valid data at the first write access).

20.7.3 How to check that the watchdog reset has been generated

If the watchdog reset flag (RSRR.WDOG) is set to "1", the watchdog reset has been generated.

20.7.4 How is the watchdog stopped?

The watchdog cannot be stopped by the software.

The watchdog can be stopped with the reset (INIT, RST, HWWD reset, SWWD reset).

20.7.5 How to clear the watchdog timer (1-bit counter)

Successively writing "A5_H" and "5A_H" in the watchdog reset generation postponement register WPR causes the 1-bit counter used for detecting the watchdog to be cleared immediately after writing "5A_H". In this state, the reset timing of the watchdog can be postponed.

In addition, if the timebase timer is cleared, the 1-bit counter used for detecting the watchdog is simultaneously reset.

20.8. Caution

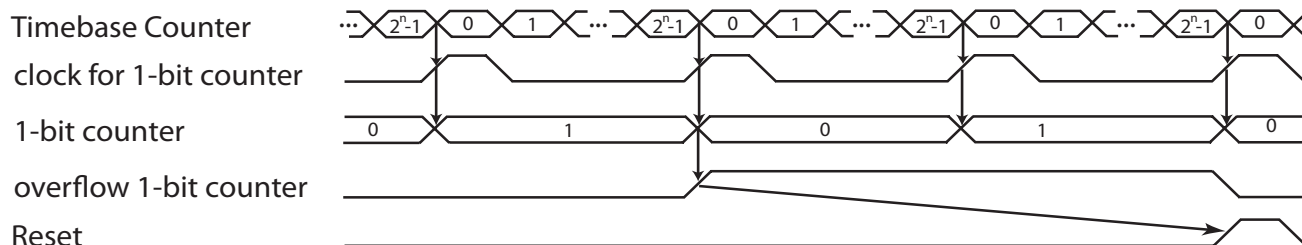
- Although the watchdog interval time corresponds to the one twice as long as the watchdog 1-bit counter, the watchdog timer clear operation only clears the 1-bit counter used for detecting the watchdog. As a result, the time margin to clear the watchdog timer is different from the interval time.

Watchdog interval time selection

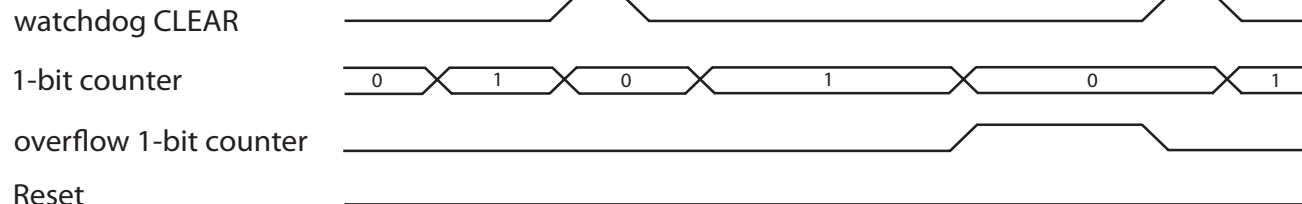
WT1,WT0	Time margin to clear the watchdog timer	Interval time during which the watchdog reset is generated
00	$\Phi \times 2^{20}$ (Initial value)	$\Phi \times 2^{20}$ to $\Phi \times 2^{21}$
01	$\Phi \times 2^{22}$	$\Phi \times 2^{22}$ to $\Phi \times 2^{23}$
10	$\Phi \times 2^{24}$	$\Phi \times 2^{24}$ to $\Phi \times 2^{25}$
11	$\Phi \times 2^{26}$	$\Phi \times 2^{26}$ to $\Phi \times 2^{27}$

Figure 20.8-1 One bit counter reset

(1) without CLEAR



(2) with CLEAR



- The watchdog timer is started once data is written in the watchdog timer control register.
- The watchdog timer control register is also the reset cause register and the status (INIT, HSTB, WDOG, ERST, SRST and LINIT) is set to "0" when it is read.
- The watchdog reset holds the oscillation stability wait time.
(Refer to "[Chapter 18 Timebase Counter \(Page No.363\)](#)".)
- When watchdog reset occurs in Main-RUN or Sub-RUN, and the Main clock is running, there will be no oscillation stabilization time.
- Refer to "[Chapter 19 Timebase Timer \(Page No.379\)](#)" for the method of clearing the timebase counter that is the count source for the watchdog timer.
- Clearing the timebase counter causes the watchdog reset timing to be postponed once.
- MB91V460A and MB91461: The watchdog timer will be retriggered / cleared automatically by DMA during D-bus access.

Chapter 21 Hardware Watchdog Timer

21.1. Overview

The hardware watchdog timer (RC oscillation based (CLKRC 100kHz)) provides a system reset if its internal timer is not cleared within the postponement duration.

21.1.1 Hardware watchdog timer

This watchdog timer starts counting after the setting initialization reset (INIT) automatically. Clearing the counter in the postponement duration is necessary to continue running an application. Otherwise if the counter is not cleared within the postponement duration, e.g. due to infinite loop in the application, this module provides a reset signal (setting initialization reset, INIT).

If the CPU is in a standby mode as described below, this watchdog timer stops:

- SLEEP state: the CPU stops, the peripherals run.
- STOP state: the CPU and the peripherals stop.
- STOP state but the RTC module and the oscillator(s) run.
- Emulation mode (emulation by EDSU or DSU4¹)

If one of the below condition occurs, the watchdog counter is cleared:

- Writing “0” to CL bit in the HWWD register
- Initialization Reset (INIT) from external pin (INITX=0) or from Clock Supervisor
- Operational Reset (Software reset, RST)
- Oscillation stops
- Transition to the SLEEP state, STOP state or “STOP with RTC running” state

Note: If HWWDE.STP_RUN ² bit is set, the watchdog continues running in SLEEP and STOP state and it is not reset at the transition to SLEEP or STOP.

21.1.2 Configuration

The Hardware Watchdog timer consists of

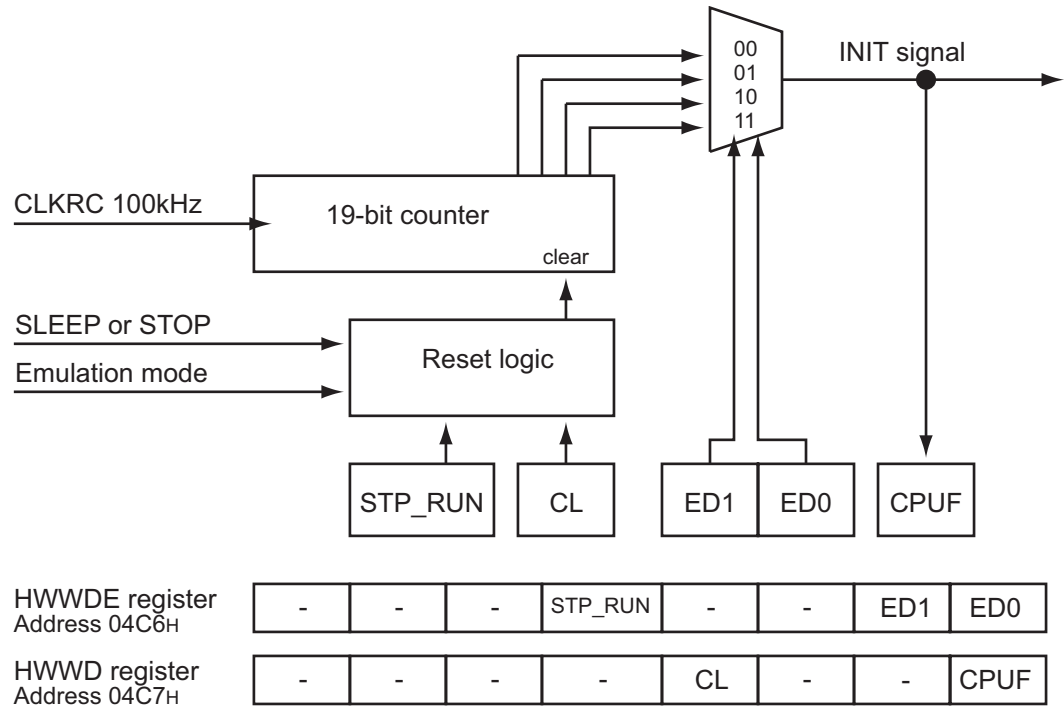
- a 19-bit counter running on the 100kHz RC clock,
- an output selector for 4 different counter outputs,
- timer control and status registers.

1. Debug Support Unit 4, available on MB91V460A and MB91FV460B

2. STP_RUN is only available on MB91FV460B, MB91F467P, MB91F467E

21.1.3 Block diagram

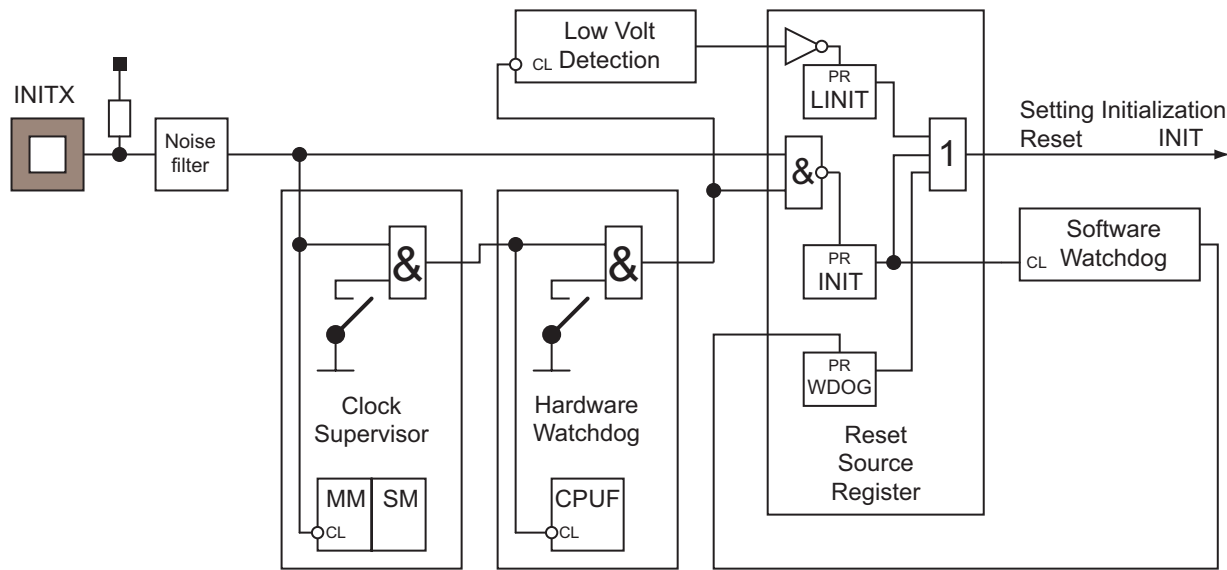
Figure 21.1-1 Block Diagram of hardware watchdog timer



21.1.4 Watchdog Reset, External Reset and Clock Supervisor Reset

The external INITX pin, the clock supervisor and the hardware watchdog built the external reset chain, which generates the Setting Initialization reset (INIT). Each module in the chain transfers the incoming reset signal to its reset output. External reset pin will clear all the modules in the chain, but the Hardware Watchdog reset will not clear the Clock Supervisor.

Figure 21.1-2 External reset / initialization chain



MB91460 Series**21.2. Register****21.2.1 Hardware watchdog timer control and status register**

Hardware watchdog timer control status register (with reset flag and clear bit).

- **HWWD: Address 04C7h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
RESV0	RESV0	RESV0	RESV1	CL	RESV0	RESV0	CPUF	
0	0	0	1	1	0	0	0	Initial value (INITX pin input or Clock Supervisor)
0	0	0	1	1	0	0	X	Initial value (Software reset)
R/W0	R/W0	R/W0	R/W1	W	R/W0	R/W0	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-5: Reserved bits. Always write “0” to these bits.
- Bit4: Reserved bit. Always write “1” to this bit.
- Bit3: CL (counter clear).

CL	Function
0	By writing ‘0’ the watchdog timer is cleared
1	Writing ‘1’ has no effect

This bit is write only. Writing “0” clears the watchdog timer.

The bit switches back to “1” automatically after the timer is cleared. Therefore the bit is always read as ‘1’.

- Bit2-1: Reserved bits. Always write “0” to these bits.
- Bit0: CPUF (CPU reset Flag).

CPUF	Function
0	Watchdog reset not triggered
1	Watchdog reset triggered (overflow of watchdog timer occurred)

This bit is initialized by external reset input (INITX) or Clock Supervisor reset, but not by internal reset.

Writing ‘0’ clears this bit, writing ‘1’ has no effect.

21.2.2 Hardware watchdog timer duration register

Hardware watchdog timer duration register (elongation of the trigger duration).

• **HWWDE: Address 04C6h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	-	STP_RUN ¹	-	-	ED1	ED0	
X	X	X	0	X	X	0	0	Initial value (INITX pin input or Clock Supervisor)
-	-	-	0	-	-	0	0	Initial value (Software reset)
RX/W0	RX/W0	RX/W0	R/W1	RX/W0	RX/W0	R/W	R/W	Attribute

1. STP_RUN is available on MB91FV460B, MB91F467P and MB91F467E only.
On other devices, always write "0" to this bit.

(See "Meaning of Bit Attribute Symbols (Page No.15)" for details of the attributes.)

- Bit7-5: Reserved bits. Always write 0 to these bits.
- Bit4: STP_RUN¹ (Run in SLEEP/STOP mode):
 - STP_RUN = 1 enables that the Hardware Watchdog continues running in SLEEP and STOP mode. The RC Oscillator will continue operation in SLEEP and STOP too, independent of CSVCR.RCE setting.
 - STP_RUN = 0 (default) the the Hardware Watchdog is cleared in SLEEP and STOP mode.
 - STP_RUN can be set by CPU only once after INIT or RST
 - STP_RUN is cleared by software reset (RST), but it cannot be cleared by the CPU
- Bit3-2: Reserved bits. Always write 0 to these bits.
- Bit1-0: ED (Elongate watchdog duration).

ED1-0	Function
00	The watchdog period is 2 ¹⁶ CLKRC cycles [initial setting]
01	The watchdog period is 2 ¹⁷ CLKRC cycles * ¹)
10	The watchdog period is 2 ¹⁸ CLKRC cycles * ¹)
11	The watchdog period is 2 ¹⁹ CLKRC cycles * ¹)

1. This setting is not available on MB91V460A.

1. STP_RUN is available on MB91FV460B, MB91F467P and MB91F467E only.
On other devices, always write "0" to this bit.

21.3. Functions

If the watchdog timer is not cleared periodically, a setting initialization reset (INIT) occurs.

21.3.1 Function of the hardware watchdog timer

After releasing INITX the hardware watchdog timer starts immediately without stabilization time. If the timer is not cleared periodically, setting initialization (INIT) reset occurs.

21.3.2 Period of the hardware watchdog timer

The timer width is 19-bit. Since the RC oscillator is used as clock source of the hardware watchdog timer, the duration of the timer deviates with the RC oscillator accuracy:

	ED1-0	Min.	Typ.	Max.
RC oscillation cycle (μ s)		5	10	20
Watchdog term (ms)	00	327.68	655.36	1310.72
	01	655.36	1310.72	2621.44
	10	1310.72	2621.44	5242.88
	11	2621.44	5242.88	10485.76

Note: The watchdog duration elongation (ED1-0) setting is not available on MB91V460A. In that case ED1-0 is always "00".

21.3.3 Running the hardware watchdog in SLEEP/STOP mode

The bit HWWDE.STP_RUN enables running the hardware watchdog in SLEEP and STOP mode. The RC oscillator is enabled in SLEEP and STOP by hardware, independent of CSVCR.RCE (RC enable) setting.

The STP_RUN feature is available on MB91FV460B, MB91F467P and MB91F467E only.

21.4. Caution

● Software disabling is not possible

The watchdog timer starts counting immediately after reset (release of INITX or RST). Software cannot stop the counting.

● Hardware disabling is only possible on the evaluation device MB91V460A, MB91FV460B

The watchdog timer can be permanently disabled by setting the corresponding jumper of the evaluation board (this is not possible on flash devices with this watchdog timer). So always ensure correct configuration of the evaluation system to reflect the behaviour of the flash device.

● Postponement of reset

In order to postpone the watchdog reset, the clearing of the watchdog timer is necessary. Whenever the CL bit of register is set to '0' (there is no minimum writing limitation), the timer is cleared and the occurrence of reset is postponed. Just writing to the register without setting CL to '0' does not clear the timer.

● Timer stop and clear

In modes where the CPU does not work (SLEEP state, STOP state or STOP with RTC active state), the timer is cleared first then the counting is stopped. See also [21.3.3 Running the hardware watchdog in SLEEP/STOP mode \(Page No.403\)](#).

● During DMA transfer

During DMA transfer between D-bus modules, the writing '0' to CL bit is not possible. Thus, if the transfer time is more than 328ms (calculated from the fastest frequency of the RC oscillator as minimum period), a reset occurs.

● Duration setting

Unlike on MB91V460A, it is possible on all devices to elongate the duration of the watchdog reset.

● CLKRC frequency

Unlike on MB91V460A, it is possible on all devices to change the CLKRC frequency to 2MHz. Even though the watchdog timer is always operated with a frequency of 100kHz (10us) typical.

● Difference between watchdog reset, external reset and clock supervisor reset

In case of a clock supervisor reset (the device is running on CLKRC then), it is necessary to have an external reset to start the device on CLKMAIN again. A watchdog reset does not have this effect.

External reset pin (INITX), Clock Supervisor and Hardware Watchdog build a "reset chain", see [21.1.4 Watchdog Reset, External Reset and Clock Supervisor Reset \(Page No.400\)](#).

Each module in the chain transfers the incoming reset signal to its reset output.

External reset pin or Power-On will clear all the modules in the chain, but the Hardware Watchdog reset will not clear the Clock Supervisor.

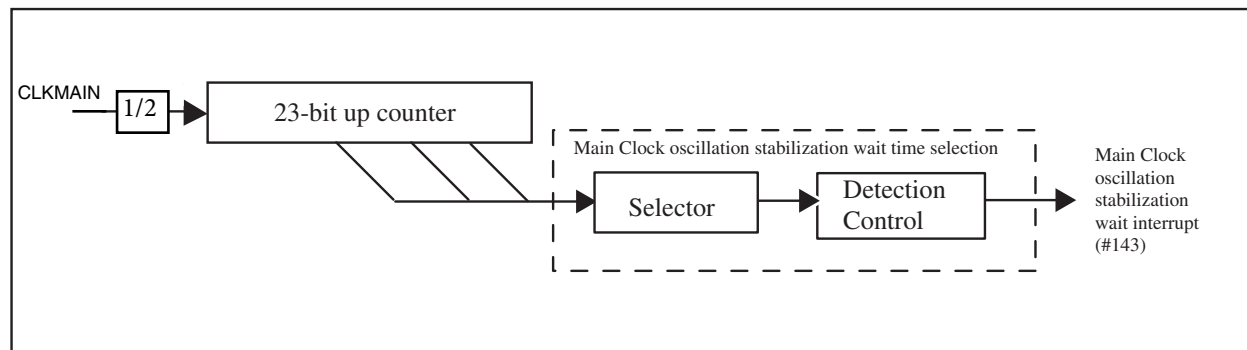
Chapter 22 Main Oscillation Stabilization Timer

22.1. Overview

The Main clock oscillation stabilization timer is a 23-bit counter that counts the CLKMAIN. This timer does not affect the selection of clock source operated by MCU/dividing setting.

This timer is mainly used for acquiring Main clock oscillation stability wait time to resume Main clock oscillation after the Main clock oscillation has been stopped (OSCCR.OSCDS1=1) while the Sub clock is being operated. In addition, this timer is best suited for interval timers or system clocks for real time OS.

Figure 22.1-1 Counter used to generate Main clock oscillation stabilization wait interrupt



22.2. Features

- Type : 23-bit Free-Run counter
- Quantity : 1
- Clock source: Main clock (source oscillation) --- Period = $2/\text{CLKMAIN}$
- Interval time: 3 types
 $\text{Period} = 2 \times 2^{12}/\text{CLKMAIN}, 2 \times 2^{17}/\text{CLKMAIN}, 2 \times 2^{23}/\text{CLKMAIN},$
 (2.0ms, 65.5ms, 4.2s / CLKMAIN 4MHz)
- Cause of timer clear: (Software, overflow, reset (INIT))
- Operation start/stop: Can be operated/stopped by the software.
- Interrupt : Main clock oscillation stability wait interrupt (Interval interrupt)
- Count value: Cannot read/write. (Clear only)

22.3. Configuration

Figure 22.3-1 Configuration Diagram of the Main clock oscillation stability wait time selection
Main clock oscillation stabilization wait timer

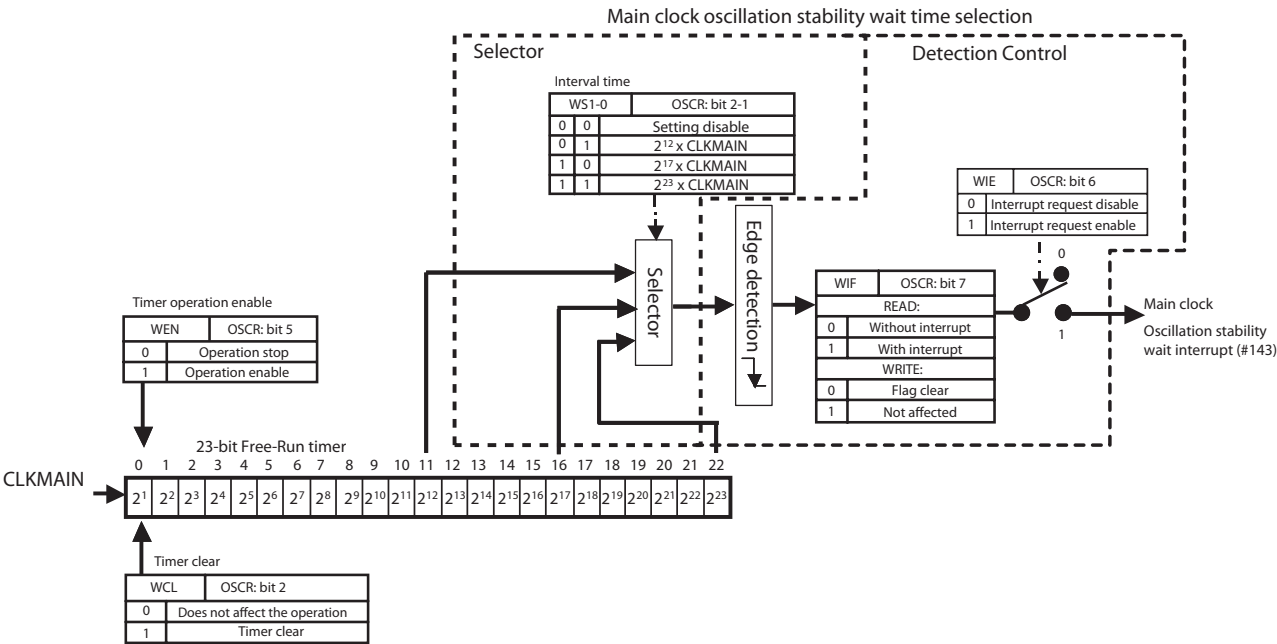


Figure 22.3-2 List of Registers

Mainclock oscillation stability wait timer									
Address	Bit	7	6	5	4	3	2	1	0
004C8H		WIF	WIE	WEN	---	---	WS1	WS0	WCL
		OSCRH							
		(Wait control status for the main clock oscillation stability)							
0047FH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0
		ICR63							
		(Interrupt level)							
Address									
0FFDC0H		32Bits							
		(Interrupt vector #143)							

*Refer to Chapter "INTERRUPT CONTROL" for the IC register and the interrupt vector.

Note: Refer to “Chapter 24 Interrupt Control (Page No.429)” for the ICR register and the interrupt vector.

22.4. Register

22.4.1 OSCRH: Control Register for the Main Clock Oscillation Stability Wait Timer

This register is used to select the interval time, clear the timer, control the interrupt, control the timer such as stop, and confirm the state of the timer.

- **OSCRH: Address 04C8h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
WIF	WIE	WEN	–	–	WS1	WS0	WCL	
0	0	0	X	X	0	0	1	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R(RM1),W	R/W	R/W	RX/W0	RX/W0	R/W	R/W	R1,W	Attribute

(For the attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- Bit7: Timer interrupt flag

WIF	Read Operation	Write Operation
0	No interrupt	Clears interrupt flag
1	With interrupt	Writing does not affect operation

- The timer interrupt flag is set to “1” at the falling edge of the selected interval period output.

- Bit6: Interrupt request enable

WIE	Operation
0	Interrupt request disable
1	Interrupt request enable

- If the timer interrupt flag (WIF) is set to “1” while the interrupt request enable (WIE) is “1” an interrupt request is immediately generated.

- Bit5: Timer operation enable

WEN	Operation
0	Stops timer operation
1	Enables timer operation

- Bit4-3: Reserved bit Be sure to write “0”. The read value is “0”.
- Bit2-1: Interval period selection

WS1	WS0	Interval period (At 4MHz)
0	0	Setting prohibited
0	1	$2 \times 2^{12} \times \text{CLKMAIN}$ (2.0ms)
1	0	$2 \times 2^{17} \times \text{CLKMAIN}$ (65.5ms)
1	1	$2 \times 2^{23} \times \text{CLKMAIN}$ (4.2s)

- The reset does not initialize. Be sure to set it after the startup.

- Bit0: Timer clear

WCL	Operation
0	Clears the Main clock oscillation stability wait timer
1	Writing does not affect operation

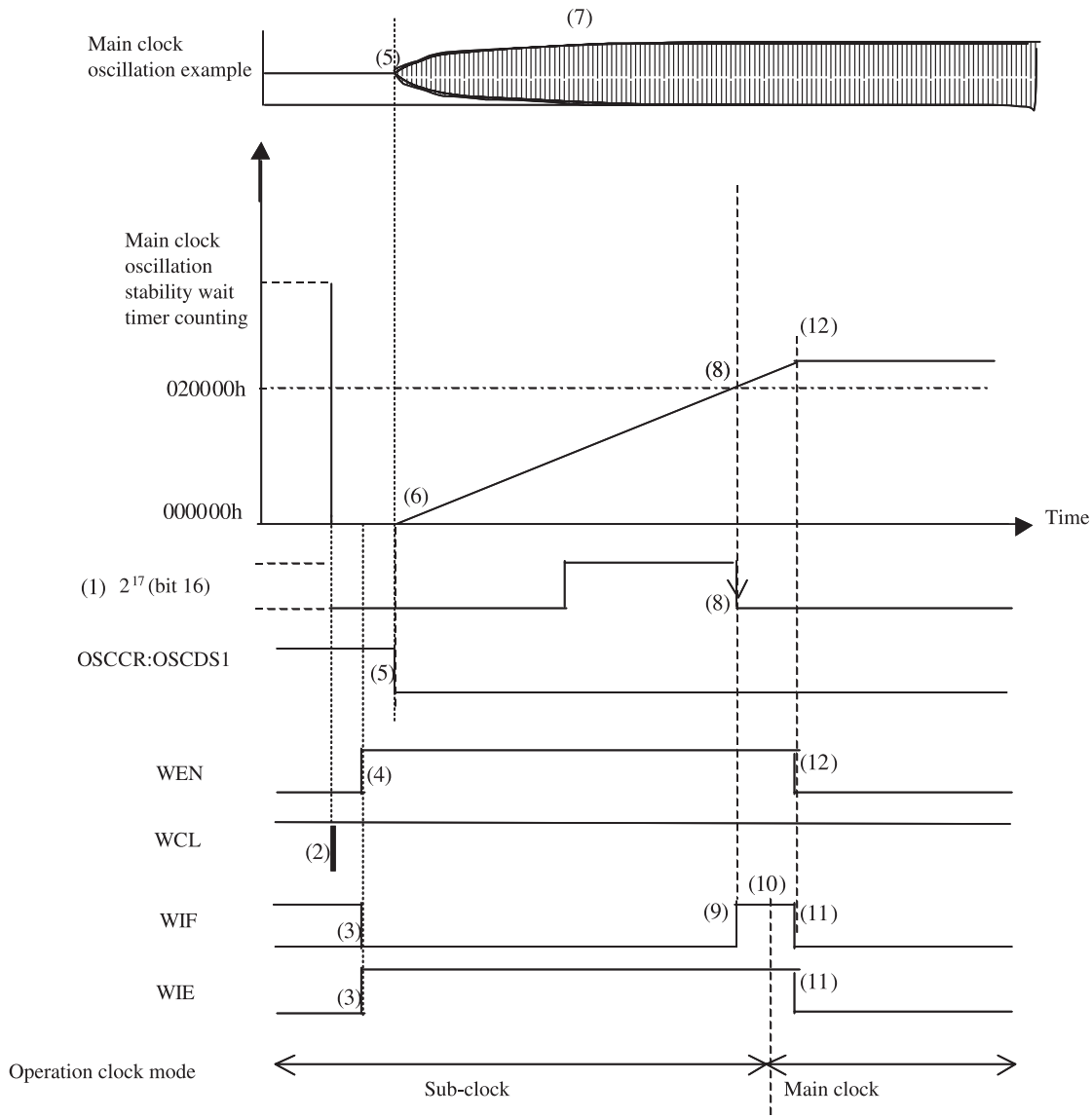
- The timer is also cleared by INITX terminal input and watchdog reset.
- About WCL, please see also section [22.8.1 Timer Clear by Writing WCL="0"](#) (Page No.414) and [22.8.2 Other Cautions](#) (Page No.414).

22.5. Operation

This section describes the Main clock oscillation stability wait timer operation.

22.5.1 Main Clock Oscillation Stability Wait

Figure 22.5-1 Example operation of Main Clock Oscillation Stability Wait

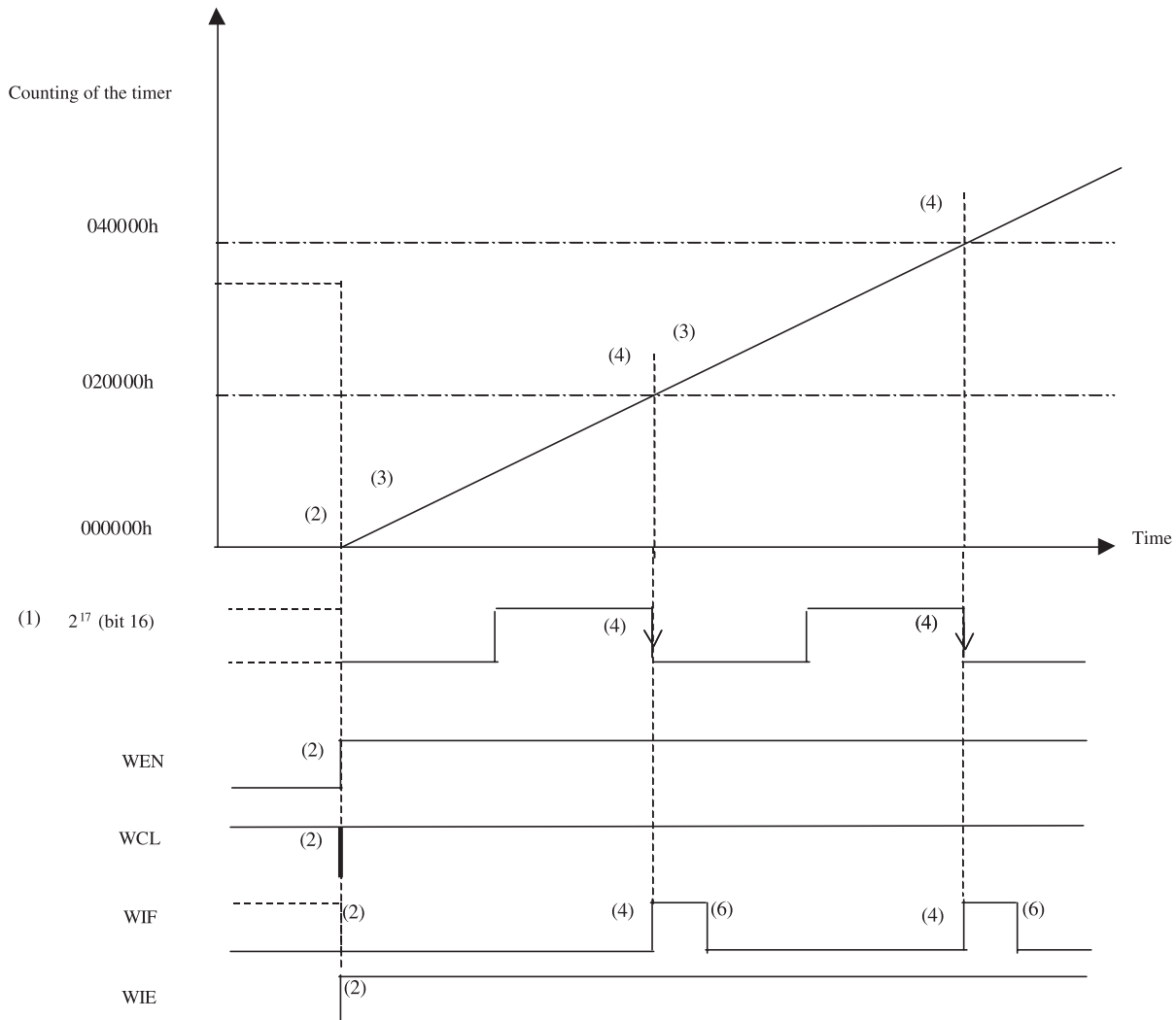


- (1) Selects the interval time. (WS[1:0]) (In this example, $2^{17}/\text{CLKMAIN}$ is selected.)
- (2) Sets timer clear (WCL="0") by the software.
- (3) Sets flag clear (WIF="0") and interrupt request enable (WIE="1") by the software.
- (4) Sets timer count enable (WEN="1") by the software.
- (5) Releases Main-STOP (OSCCR.OSCDS1="0") while the Sub clock is in operation by the software, and starts Main clock oscillation.
- (6) Starts counting (The timer counts up using the Main clock (source oscillation).)
- (7) Stabilizes the Main clock oscillation.
- (8) The selected interval time is used. (Detects the falling edge of the dividing 2^{17} .)

- (9) Generates a Main clock oscillation stability wait interrupt.
- (10) Processing caused by an interrupt (Software): Operation clock switching (Sub-RUN => Main-RUN)
- (11) Disables interrupt request (WIE="0") and clears interrupt flag (WIF="0").
- (12) Stops counting (WEN="0").

22.5.2 Interval Interrupt

Figure 22.5-2 Example of generating an interval interrupt



- (1) Selects the interval time (WS[1:0]). (In this example, 2¹⁷/CLKMAIN is selected.)
- (2) Clears the timer (WCL="0"), clears flags (WIF="0"), enables interrupt request (WIE="1"), enables timer count (WEN="1") by the software.
- (3) The timer counts up using the Main clock (source oscillation).
- (4) Generates interval interrupt at the selected interval time (Falling of the dividing 2¹⁷).
- (5) Processing caused by an interrupt (Software): Clears interrupt flag (WIF="0").
- (6) Repeats Items (3) to (5)

22.6. Setting

Figure 22.6-1 Settings Required for Using the Main Clock Oscillation Stability Wait Timer

Setting	Setting register	Setting method*
Setting interval time	Main clock oscillation stability wait timer control register (OSCRH)	7.1
Count clear		7.4
Counting operation start		7.3

*: Refer to the number for more information on the setting method.

Figure 22.6-2 Settings Required for Enabling the Main Clock Oscillation Stability Wait Timer Interrupt

Setting	Setting register	Setting method*
Sets the Main clock oscillation stability wait timer interrupt vector and Sets Free-Run timer interrupt level	Refer to “ Chapter 24 Interrupt Control (Page No.429) ”.	7.5
Sets the Main clock oscillation stability wait timer interrupt Clears interrupt flag Enables interrupt request	The Main clock oscillation stability wait timer control register (OSCRH)	7.7

*: Refer to the number for more information on the setting method.

Figure 22.6-3 Settings Required for Stopping the Main Clock Oscillation Stability Wait Timer

Setting	Setting register	Setting method*
Sets the Main clock oscillation stability wait timer stop	The Main clock oscillation stability wait timer control register (OSCRH)	7.8

*: Refer to the number for more information on the setting method.

22.7. Q & A

22.7.1 What are the types of interval time (wait time) and how are they selected?

There are 3 types of interval time, and they are set with the interval selection bit (OSCRH.WS[0:1]).

Interval time	Count period	Interval (Wait time) Example
	Interval selection bit (WS[1:0])	At CLKMAIN = 4.00MHz
To set the value to 2×2^{12} / CLKMAIN	Set the value to "01"	2.00ms
To set the value to 2×2^{17} / CLKMAIN	Set the value to "10"	65.5ms
To set the value to 2×2^{23} / CLKMAIN	Set the value to "11"	4.19 s

Note: Setting (WS[1:0]="00") is prohibited.

22.7.2 How to select the count clock

The count clock is the Main clock (source oscillation). (Cannot be selected.)

22.7.3 How is the Main clock oscillation stabilization wait timer count operation enabled/disabled?

Set/unset the timer operation enable bit (OSCRH.WEN).

Operation	Timer operation enable bit (WEN)
To stop the Main clock oscillation stabilization wait timer	Set the value to "0"
To start the Main clock oscillation stabilization wait timer	Set the value to "1"

22.7.4 How is the Main clock oscillation stabilization wait timer cleared?

One of the following methods can be used to clear the Main clock oscillation stabilization wait timer.

- Write "0" to the clear bit (OSCRH.WCL).

Operation	Clear bit (WCL)
To clear the Main clock oscillation stabilization wait timer	Write "0"

- Perform a reset.

Clear the Main clock oscillation stabilization wait timer with the operation initialization reset (INITX pin input, watchdog reset).

Note: The operation initialization reset (Software reset) holds the count of the Main clock oscillation stabilization wait timer.

- The overflow (Next count-up for "7FFFFFFh") of the Main clock oscillation stability wait timer causes the count value to be reset to "000000_H".
- About WCL, please see also section [22.8.1 Timer Clear by Writing WCL="0" \(Page No.414\)](#) and [22.8.2 Other Cautions \(Page No.414\)](#).

22.7.5 What are the interrupt-associated registers?

Setting the interrupt vector and the interrupt level of the Main clock oscillation stability wait timer.

The relationship between the interrupt level and the interrupt vector is shown in the following table.

Refer to "[Chapter 24 Interrupt Control \(Page No.429\)](#)" for the interrupt level and interrupt vector.

Interrupt vector (Default)	Interrupt level setting bit (ICR4-ICR0)
----------------------------	---

#143 Address: 0FFDC0h	Interrupt level register (ICR63) Address: 047Fh
--------------------------	--

As the interrupt flag (OSCRH.WIF) is not automatically cleared, clear it before returning from the interrupt processing by the software. (Write “0” in the WIF bit.)

22.7.6 What are the types of interrupt?

There is one type of interrupt called the Main clock oscillation stability wait timer interrupt.
(Selection is unnecessary.)

22.7.7 How is the interrupt request enabled?

Interrupt request enable and interrupt flag.

Setting the interrupt request enable is performed with the interrupt request enable bit (OSCRH.WIE).

	Interrupt request enable bit (WIE)
Interrupt request disable	Set the value to “0”
Interrupt request enable	Set the value to “1”

Clearing an interrupt is performed with the interrupt flag (OSCRH.WIF).

	Interrupt flag (WIF)
Interrupt clear	Writes “0”

22.7.8 How is the Main clock oscillation stability wait timer stopped counting?

Sets with the timer operation enable bit (OSCRH.WEN). Refer to 7.3.

In addition, if the MCU stops the Main clock while the Sub clock is being operated, the Main clock oscillation stability wait timer also stops counting.

22.8. Caution

22.8.1 Timer Clear by Writing WCL="0"

- Timer clear by writing WCL="0" is delayed on all devices (but not on MB91460E series). After writing "0" to the WCL bit, the timer is not cleared immediately, it stops counting only. To finally clear and restart the counter, a sub-subsequent access to an address within the R-Bus address area must be done. This can be done by either writing or reading another register of a resource (connected to the R-Bus) or simply read the RBSYNC address (0x03A).

Correct sequence example:

```
OSCRH_WCL = 0;           // Initiate counter clearing, counter will stop now
RBSYNC;                  // Read access to R-bus to force timer to clear and restart
```

- **Attention:** If no R-bus access follows the sequence of setting OSCRH_WCL = "0" the counter will stop only. It will clear and restart not until another R-bus access is done.
- **Note on MB91460E series:** The above described behavior is different for MB91460E series devices. Writing WCL = "0" will immediately clear and restart the timer. This means on MB91F467E the Osci Stabi Timer Interrupts may appear earlier than on other devices running the same software.

22.8.2 Other Cautions

- To wait until the Main clock oscillation stability is attained while the Sub clock is in operation, it is necessary to acquire wait time using the Main clock oscillation stability wait timer.
(An unstable clock may be supplied to the entire device, and normal operation is not guaranteed if the MCU operation mode is switched from the Sub-RUN to the Main-RUN mode without waiting until the Main clock oscillation becomes stable.)
- The value for the oscillation stability wait time is an estimated value because the oscillation period of the Main clock oscillation is unstable for the beginning immediately after the oscillation has started.
- If the Main clock oscillation stops, the Main clock oscillation stability wait interrupt (interval interrupt) is not generated either because the Main clock oscillation stability wait timer stops. The Main clock oscillation should be enabled for processing that uses the Main clock oscillation stability wait interrupt (interval interrupt).
- The flag is set to "1" (flag setting preference) if the timer interrupt (WIF="1") and the writing operation where "0" is written by software in the flag occur simultaneously.
- The Main clock oscillation stability wait timer is counted up with the Main clock. As a result, in the following state, the counting of the timer used to stop the Main clock oscillation also stops.
 - If the timer operation enable bit (OSCRH.WEN) is "0", the timer stops counting.
 - If the Main clock is stopped in the STOP state (STCR.OSCD1="1"), the timer stops counting from the moment the STOP state is activated.
 - If the Main clock oscillation is stopped (OSCCR.OSCDS1="1") during Sub clock operation, the timer stops while the Sub clock is in operation.
- If the interrupt request should be enabled (WIE="1") after the reset is released, and the interval time to be modified, be sure to simultaneously set the interrupt flag (WIF) and the clear bit (WCL) to "0" beforehand.
- The timer interrupt flag (WIF), timer interrupt request enable bit (WIE), timer enable bit (WEN) and timer

clear bit (WCL) are initialized using the setting initialization reset (INITX pin input, watchdog reset).

- Be sure to set the interval selection bit (WS[1:0]) after startup (after setting initialization reset) by the software.
- The Main clock oscillation stability wait timer control register should be initialized (to set the initial value) only with the setting initialization reset (INITX pin input, watchdog reset) because the software reset does not initialize the register and the current value is held.
- If the counter clear (WPCR.WCL="0") and the overflow for the selected bit occur simultaneously, the interrupt flag (WIF) is not set to "1".

Chapter 23 Sub Oscillation Stabilization Timer

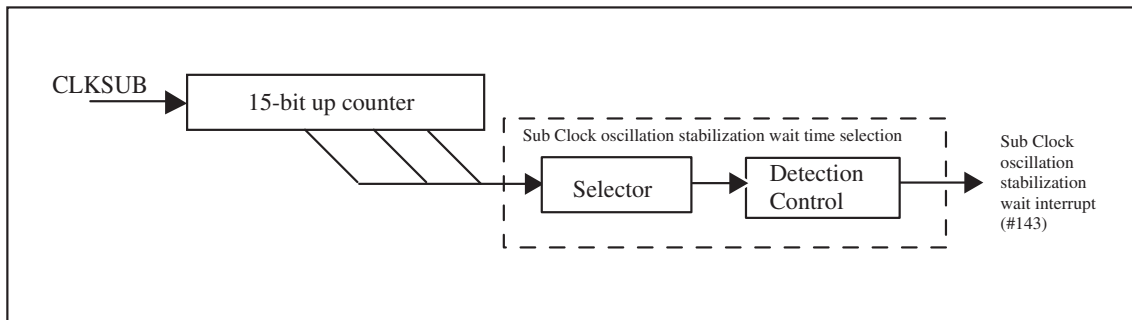
23.1. Overview

The sub oscillation stabilization timer is a 15-bit counter that is counted up with the Sub clock. This timer does not affect the selection/dividing setting of the MCU operating clock.

This timer is used to acquire Sub clock oscillation stability wait time if the Sub clock oscillation is resumed mainly when the Sub clock oscillation is stopped while the Main clock is in operation.

This timer is can be used for acquiring Sub clock oscillation stability wait time to resume Sub clock oscillation after the Sub clock oscillation has been stopped (OSCCR.OSCDS2=1) while the RC oscillator is being operated.

Figure 23.1-1 Counter used to generate Sub clock oscillation stabilization wait interrupt



23.2. Features

- Type : 15-bit Free-Run counter
- Quantity : 1
- Clock source: CLKSUB (source oscillation) --- Period = $1/\text{CLKSUB} = 1/32.768\text{kHz}$
- Interval time: 4 types
 $\text{Period} = 2^{10}/\text{CLKSUB}, 2^{13}/\text{CLKSUB}, 2^{14}/\text{CLKSUB}, 2^{15}/\text{CLKSUB},$
 (31.25ms, 0.25s, 0.50s, 1.00s)
- Timer clear cause: (Software, overflow, reset (INIT))
- Interrupt : clock interrupt (interval interrupt)
- Count value: Cannot read and write (Clear only)

23.3. Configuration

Figure 23.3-1 Configuration Diagram of the Sub clock oscillation stabilization timer
Sub clock oscillation stabilization wait timer

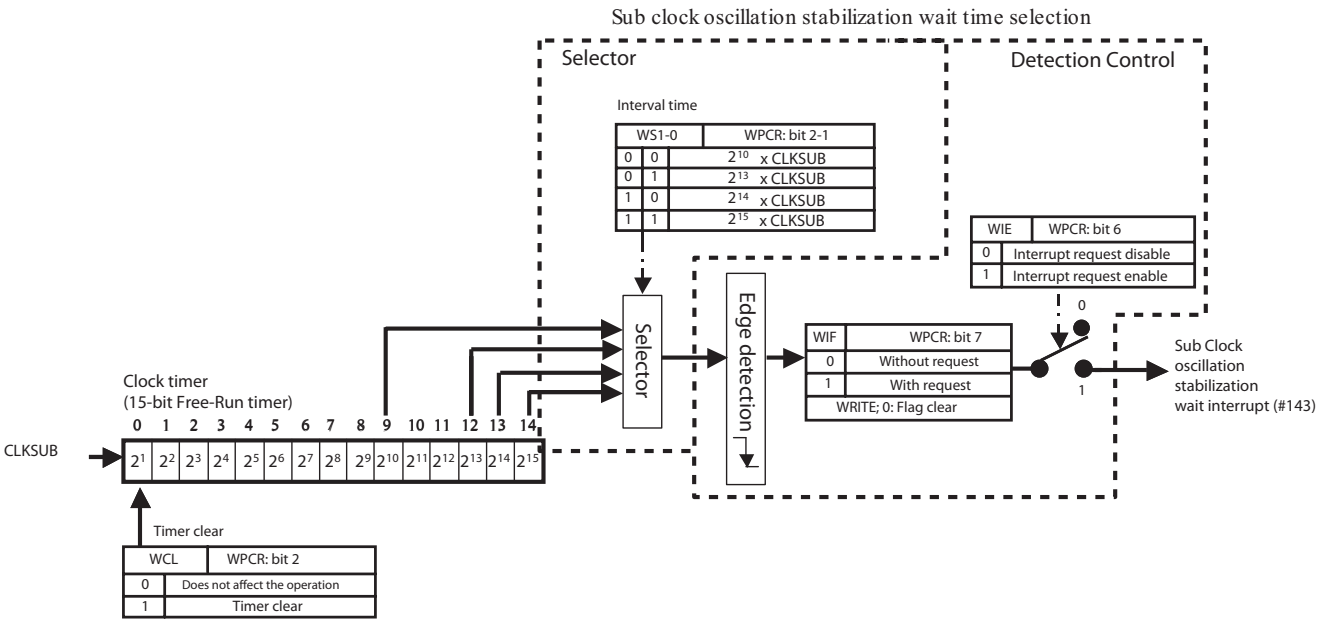


Figure 23.3-2 List of Registers

Sub clock timer									
Address	Bit	7	6	5	4	3	2	1	0
004CAH		WIF	WIE	---	---	---	WS1	WS0	WCL
		WPCRH							
		(Clock timer control status)							
0047FH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0
		ICR63							
		(Interrupt level)							
Address									
0FFDC0H	32Bits								
	(Interrupt vector #143)								

*Refer to Chapter 'INTERRUPT CONTROL' for the IC register and the interrupt vector.

Note: For the ICR register and interrupt vector, refer to “Chapter 24 Interrupt Control (Page No.429)”.

23.4. Register

23.4.1 WPCRH: Sub oscillation stabilization timer Control Register

This register is used to select interval time, clear the timer, control interrupt, control timer stop etc., and confirm the states.

- **WPCRH: Address 04CAh (Access: Byte)**

7	6	5	4	3	2	1	0	bit
WIF	WIE	—	—	—	WS1	WS0	WCL	
0	0	0	X	X	0	0	1	Initial value (At INIT)
R(RM1),W	R/W	R/W	RX/W0	RX/W0	R/W	R/W	R1,W	Attribute

(For the attributes, refer to “Meaning of Bit Attribute Symbols (Page No.15)”.)

(Refer to “23.8. Caution (Page No.426)”.)

- Bit7: Sub oscillation stabilization timer interrupt flag

WIF	Read Operation	Write Operation
0	Without interrupt	Clears the interrupt flag
1	With interrupt	Writing does not affect operation

- The sub oscillation stabilization timer interrupt flag is set to “1” at the falling edge of the selected interval period output.
- Bit6: Interrupt request enable

WIE	Operation
0	Interrupt request is disabled
1	Interrupt request is enabled

- If the sub oscillation stabilization timer interrupt flag is (WIF=“1”), and if the interrupt request enable bit (WIE) is set to “1”, an interrupt request is immediately generated.
- Bit5-3: Reserved bit. Be sure to write “0”. The read value is “0”.
- Bit2-1: Interval period selection

WS1	WS0	Interval period (CLKSUB = 32.768kHz)
0	0	$2^{10}/\text{CLKSUB}$ (31.25ms)
0	1	$2^{13}/\text{CLKSUB}$ (0.25s)
1	0	$2^{14}/\text{CLKSUB}$ (0.50s)
1	1	$2^{15}/\text{CLKSUB}$ (1.00s)

- Bit0: Timer clear

WCL	Operation
0	Clears the sub oscillation stabilization timer.
1	Writing does not affect write operation.

- The timer is also cleared by INITX terminal input and watchdog reset.

Notes 1: Initial value can be set using the setting initialization reset (INITX pin input, watchdog reset), but the operation initialization reset (Software reset) holds the current value instead of initializing it.

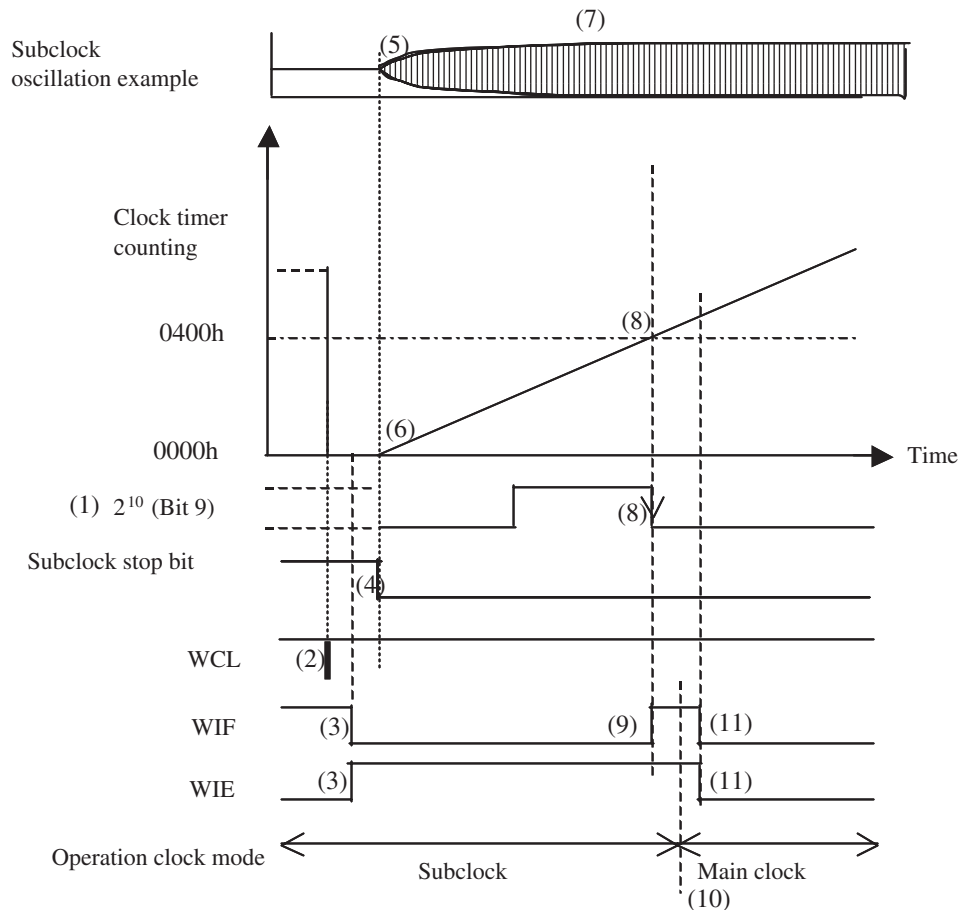
2: If the interrupt request enable (WIE=“1”) is set, and the interval period selection (WS[1:0]) after canceling the reset, be sure to simultaneously set the timer interrupt flag (WIF) and the timer clear (WCL) “0”.

3: About WCL, please see also section 23.8.1 Timer Clear by Writing WCL=“0” (Page No.426).

23.5. Operation

23.5.1 Sub clock Oscillation Stability Wait Interrupt

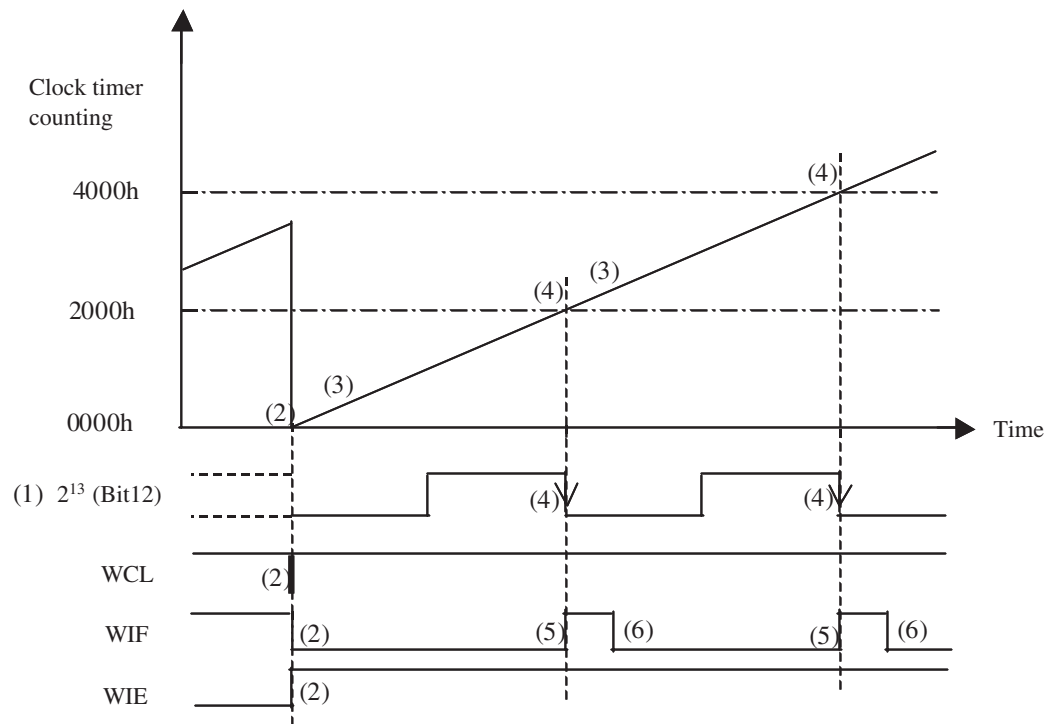
Figure 23.5-1 Example of generating a Sub clock Oscillation Stability Wait Interrupt



- (1) Selects the interval (WS[1:0]) (In this example, $2^{10}/\text{CLKSUB}$ is selected.)
- (2) Sets the timer so that it is cleared (WCL="0") by software.
- (3) Sets the flag clear (WIF="0") and the interrupt request enable (WIE="1") by software.
- (4) Sets the Sub-STOP release (OSCCR.OSCDS1="0") while the Sub clock is in operation by software.
- (5) The Sub clock oscillation starts.
- (6) Counts up with the Sub clock (source oscillation).
- (7) Make the Sub clock oscillation stable.
- (8) Makes the interval time be the selected time. (Detects the falling of 2^{10} dividing.)
- (9) If the flag (WIF) becomes "1", the Sub clock oscillation stability wait interrupt request is generated.
- (10) Processing cause by an interrupt (Software): Switching the operation clock (Sub-RUN => Main-RUN)
- (11) Interrupt request disable (WIE="0") and the interrupt flag clear (WIF="0").

23.5.2 Interval Interrupt (Clock Interrupt)

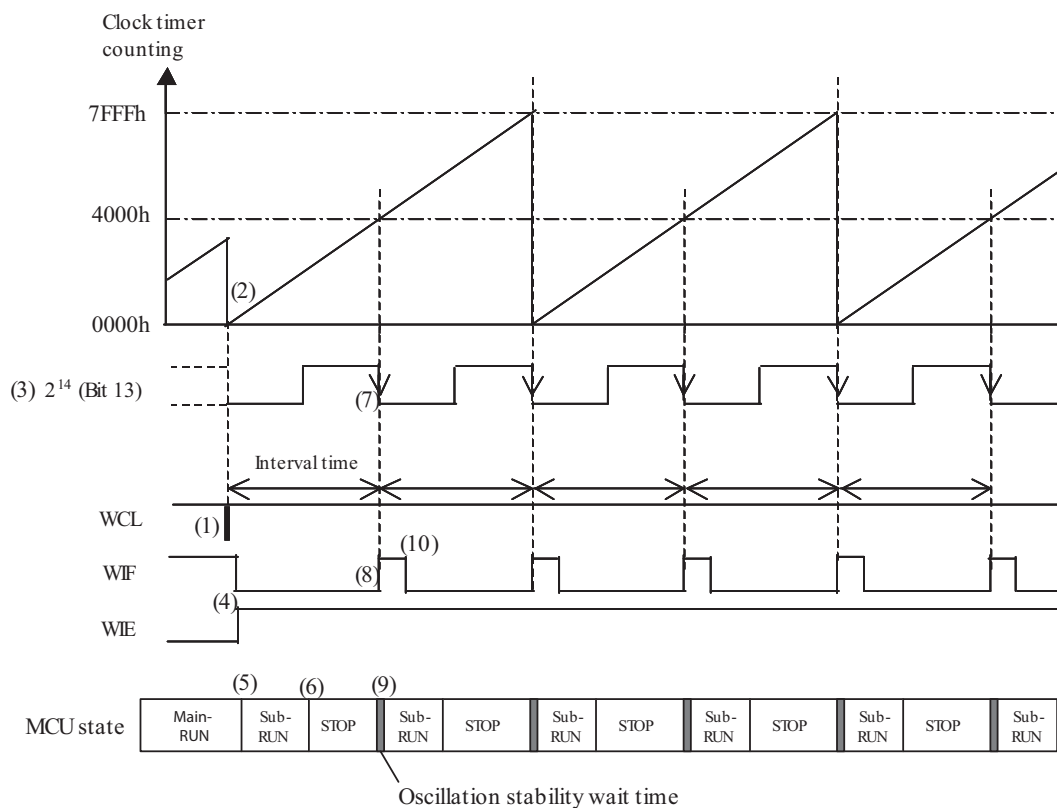
Figure 23.5-2 Example of generating an Interval Interrupt (Clock Interrupt)



- (1) Selects the interval time. (WS[1:0]) (In this example, $2^{13}/\text{CLKSUB}$ is selected.)
- (2) Sets the timer clear (WCL="0"), flag clear (WIF="0") and interrupt request enable (WIE="1") by the software.
- (3) The timer counts up with the Sub clock (Source oscillation).
- (4) Makes the interval time be the selected time. (Detects the fall of 2^{13} .)
- (5) If the flag (WIF) is set to "1", interval interrupt request (Clock interrupt request) is generated.
- (6) Processing caused by an interrupt (Software): The interrupt flag clear (WIF="0")
(Arbitrary processing such as clock counting)
- (7) Repeats Items (3) to (6).

23.5.3 Returning from the STOP state due to Interval Operation (Clock Interrupt)

Figure 23.5-3 Returning from the STOP state due to Interval Operation (Clock Interrupt)



- (1) The sub oscillation stabilization timer is cleared by software. (Writes "0" to WCL.)
- (2) Counts up the sub oscillation stabilization timer with the Sub clock.
- (3) Selects the interval time. (In this example, 0.5 second: Selects WS[1:0]="10".)
- (4) Sets the flag clear (WIF="0") and sub oscillation stabilization timer interrupt enable (WIE="1") by the software.
- (5) Switches the MCU operation from the Main-RUN to Sub-RUN.
- (6) Switches to the STOP state.
- (7) Makes the interval time be the selected time. (0.5 second)
- (8) The interrupt flag (WIF) is set to "1".
- (9) As the interrupt request is enabled (WIE="1"), returns from the STOP state to Sub-RUN.
- (10) Clears the interrupt flag by software. (Writes "0" to the WIF.)
- (11) Repeats Items (6) from (10).

23.6. Setting

Table 23.6-1 Settings Required for Using the Sub oscillation stabilization timer

Setting	Setting register	Setting method*
Setting the interval time	Sub oscillation stabilization timer control register (WPCRH)	Refer to 7.1.
Count clear		Refer to 7.4.

*: Refer to the number for more information on the setting method.

Table 23.6-2 Items Required for Enabling the Sub oscillation stabilization timer Interrupt

Setting	Setting register	Setting method*
Setting the interrupt vector and the Free-Run timer level of the sub oscillation stabilization timer	Refer to “ Chapter 24 Interrupt Control (Page No.429) ”.	Refer to 7.5.
Setting the sub oscillation stabilization timer interrupt Clearing the interrupt flag Enabling the interrupt request	Sub oscillation stabilization timer control register (WPCRH)	Refer to 7.7.

*: Refer to the number for more information on the setting method.

23.7. Q & A

23.7.1 What are the types of interval time (wait time) and how are they selected?

There are three types of interval time, and they are set with the interval selection bit (WPCR.H.WS[1:0]).

Interval time	Count period	Interval (Wait time) Example
	Interval selection bit (WS[1:0])	CLKSUB = 32.768kHz
To set the interval time to $2^{10}/\text{CLKSUB}$	Set the value to "00".	31.25ms
To set the interval time to $2^{13}/\text{CLKSUB}$	Set the value to "01".	0.25s
To set the interval time to $2^{14}/\text{CLKSUB}$	Set the value to "10".	0.50s
To set the interval time to $2^{15}/\text{CLKSUB}$	Set the value to "11".	1.00s

23.7.2 How is the count clock selected?

The count clock is the Sub clock (source oscillation).

23.7.3 How is the Sub clock oscillation stabilization wait timer cleared?

One of the following methods can be used to clear the Sub clock oscillation stabilization wait timer.

- Write "0" to the clear bit (WPCR.H.WCL).

Operation	Clear bit (WCL)
To clear the Sub clock oscillation stabilization wait timer	Write "0"

- Perform a reset.

Clear the Sub clock oscillation stabilization wait timer with the operation initialization reset (INITX pin input, watchdog reset).

Note: The operation initialization reset (Software reset) holds the count of the Sub clock oscillation stabilization wait timer.

- The overflow (Next count-up for "7FFFh") of the Sub clock oscillation stability wait timer causes the count value to be reset to "0000_H".
- About WCL, please see also section [23.8.1 Timer Clear by Writing WCL="0"](#) (Page No.426).

23.7.4 What are interrupt-associated registers?

Setting the interrupt vector and the interrupt level of the Sub clock oscillation stability wait timer.

The relationship between the interrupt level and the interrupt vector is shown in the following table.

Refer to "[Chapter 24 Interrupt Control \(Page No.429\)](#)" for more information on the interrupt level and the interrupt vector.

Interrupt vector (Default)	Interrupt level setting bit (ICR[4:0])
#143 Address: 0FFDC0h	Interrupt level register (ICR63) Address: 047Fh

As the interrupt flag (WPCR.H.WIF) is not automatically cleared, clear it before returning from the interrupt processing by the software. (Write "0" to the WIF bit.)

23.7.5 What are the types of interrupt?

There is one type for the interrupt, and it is generated with the interval time (Sub clock oscillation stability wait).

23.7.6 How is the interrupt request enabled?

The interrupt request enable and the interrupt flag.

The interrupt request is enabled through setting the interrupt request enable bit (WPCR.H.WIE).

	Interrupt request enable bit (WIE)
Interrupt request disable	Set the value to "0"
Interrupt request enable	Set the value to "1"

The interrupt request is cleared with the interrupt flag (WPCR.H.WIF).

	Interrupt flag (WIF)
Interrupt flag clear	Writes "0"

23.7.7 How is the Sub clock oscillation stability wait timer stopped counting?

The timer is always running as long as the Sub clock is available.

In addition, if the MCU stops the Sub clock while the Main clock is being operated, the Sub clock oscillation stability wait timer also stops counting.

23.8. Caution

23.8.1 Timer Clear by Writing WCL="0"

- Timer clear by writing WCL="0" is delayed on all devices (but not on MB91460E series). After writing "0" to the WCL bit, the timer is not cleared immediately, it stops counting only. To finally clear and restart the counter, a sub-sequent access to an address within the R-Bus address area must be done. This can be done by either writing or reading another register of a resource (connected to the R-Bus) or simply read the RBSYNC address (0x03A).

Correct sequence example:

```
WPCRH_WCL = 0;           // Initiate counter clearing, counter will stop now
RBSYNC;                  // Read access to R-bus to force timer to clear and restart
```

- **Attention:** If no R-bus access follows the sequence of setting WPCRH_WCL = "0" the counter will stop only. It will clear and restart not until another R-bus access is done.
- **Note on MB91460E series:** The above described behavior is different for MB91460E series devices. Writing WCL = "0" will immediately clear and restart the timer. This means on MB91F467E the Osci Stabi Timer Interrupts may appear earlier than on other devices running the same software.

23.8.2 Other Cautions

- If the setting (WIF="1") of the timer interrupt flag and a write "0" operation to the flag by the software occur simultaneously, the flag is set to "1".
- If the interrupt request is enabled (WIE="1") after defeating a reset, and if the interval time is changed, be sure to simultaneously set "0" to the interrupt request enable flag (WIF) and the clear bit (WCL).
- Read-modify-write
The interrupt flag (WIF) is always read as "1" with the Read-modify-write.
- The setting initialization reset (INITX terminal input, watchdog reset) initializes the values of the timer interrupt flag (WIF), timer interrupt request enable bit (WIE) and timer clear bit (WCL) to "0", but cannot initialize the interval period selection bit (WS[1:0]). Be sure to set it by the software.
- Setting the initial value of the sub oscillation stabilization timer control register is possible using the initialization reset (INITX terminal input, watchdog reset), but the operation initialization reset (Software reset) holds the current value instead of initializing the value of the sub oscillation stabilization timer control register.
- The value for the oscillation stability wait time is an estimated value because the oscillation period of the Main clock oscillation is unstable for the beginning immediately after the oscillation has started.
- An unstable clock may be supplied to the entire device, and normal operation is not guaranteed if the Sub clock is made to oscillate starting from Sub clock stopped state, and if the MCU operation mode is switched from the Main-RUN to the Sub-RUN state without waiting until the Sub clock oscillation becomes stable. Be sure to acquire the Sub clock oscillation stability wait time using the sub oscillation stabilization timer, etc. (If the Main clock is selected as the clock source, the oscillation stability wait time for the Sub clock may not be acquired.)
- The value for the oscillation stability wait time is an estimated value because the oscillation period of the Sub

clock is unstable for the beginning immediately after it has started.

- As the sub oscillation stabilization timer stops while the Sub clock stops oscillating, a clock interrupt (interval interrupt) is not generated either. If processing using the clock interrupt (interval interrupt) is performed, enable the Sub clock oscillation. (Do not stop the Sub clock oscillation).
- The sub oscillation stabilization timer counts up with the Sub clock. As a result, the timer stops counting because the Sub clock stops oscillating under the following conditions.
 - If the Sub clock is set that it stops in the STOP state (Sub clock oscillation enable bit* =“1”), and then the mode is switched to the STOP state, the sub oscillation stabilization timer stops counting while in the STOP state.
 - If the sub oscillation stabilization timer should continue counting while in STOP state, set the Sub clock oscillation enable bit to “0” before switching to the STOP state.
 - If the Sub clock stop bit =“1” while in the Sub clock, and if the Sub clock is specified so that it stops oscillating while the Sub clock is in operation, the sub oscillation stabilization timer stops, too, while the Sub clock is in operation.

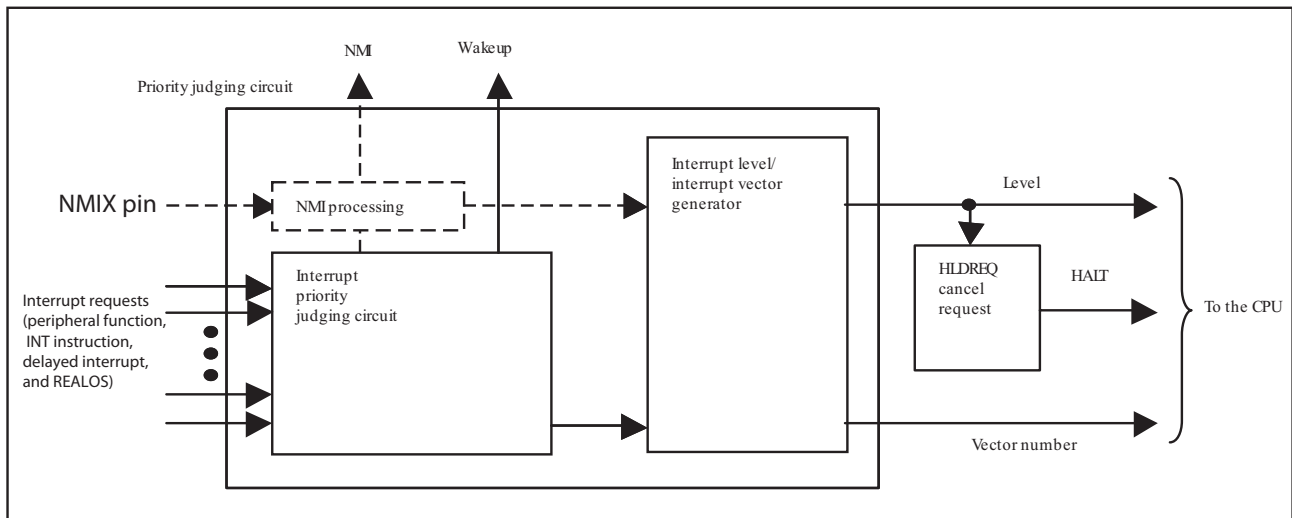
MB91460 Series

Chapter 24 Interrupt Control

24.1. Overview

Interrupt control manages interrupt reception and arbitration.

Figure 24.1-1 Block diagram of Interrupt Control



24.2. Features

- Functions
 - Detection of interrupt requests
 - The interrupt priority is determined by interrupt level and interrupt number. The first criteria is the interrupt level. For more than one interrupt with the same interrupt level at the same time, the interrupt number is determinant.
 - The interrupt level from the event with the highest priority is propagated to the CPU.
 - The interrupt number from the event with the highest priority is propagated to the CPU.
 - Request (to the CPU) to return from STOP state by a valid interrupt (Wakeup)
- Interrupt level
 - Reserved for System : level 0 to 14
 - NMI : level 15
 - Interrupt : level 16 to 31
 - Interrupt disable : level 31
(As the interrupt level goes up, the interrupt priority goes down.)
- Number of interrupt triggers
 - NMI : 1
 - Interrupt from peripheral functions : 128
 - Delayed interrupt : 1
 - Reserved for system (for REALOS) : 2
 - INT instruction : 111

24.3. Configuration

Figure 24.3-1 Configuration Diagram of Interrupt Control

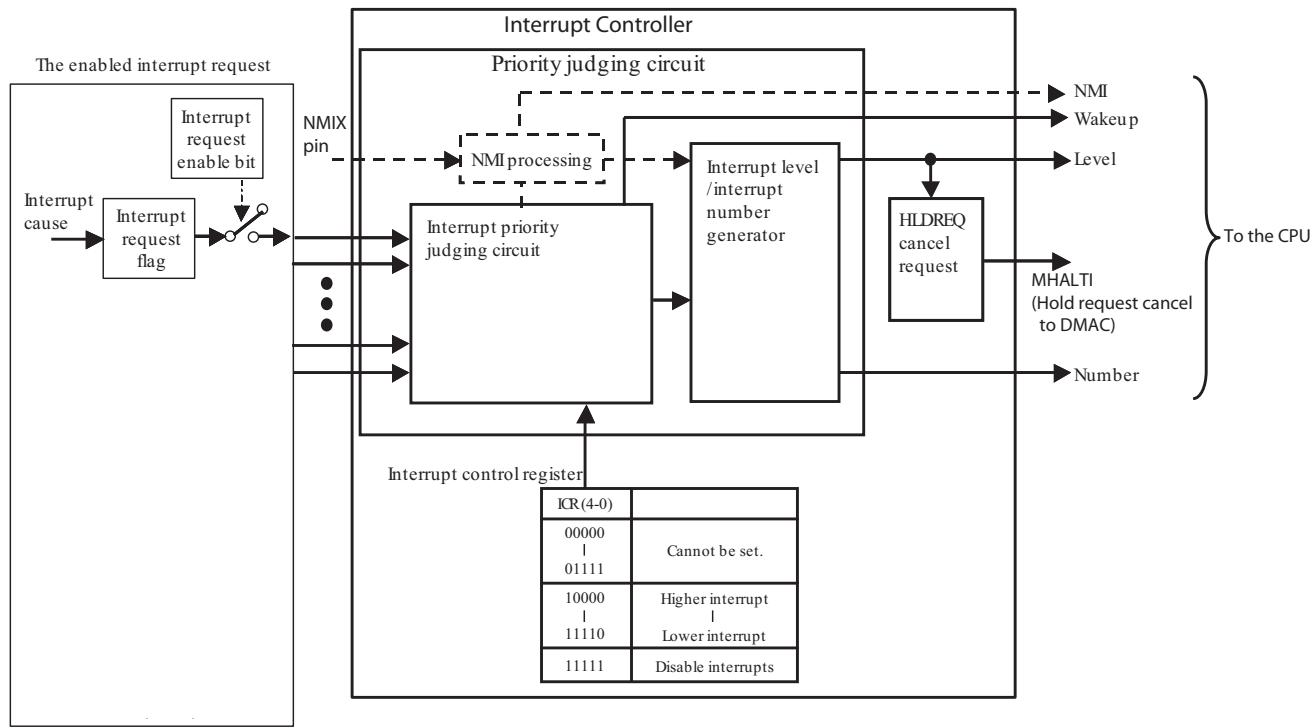
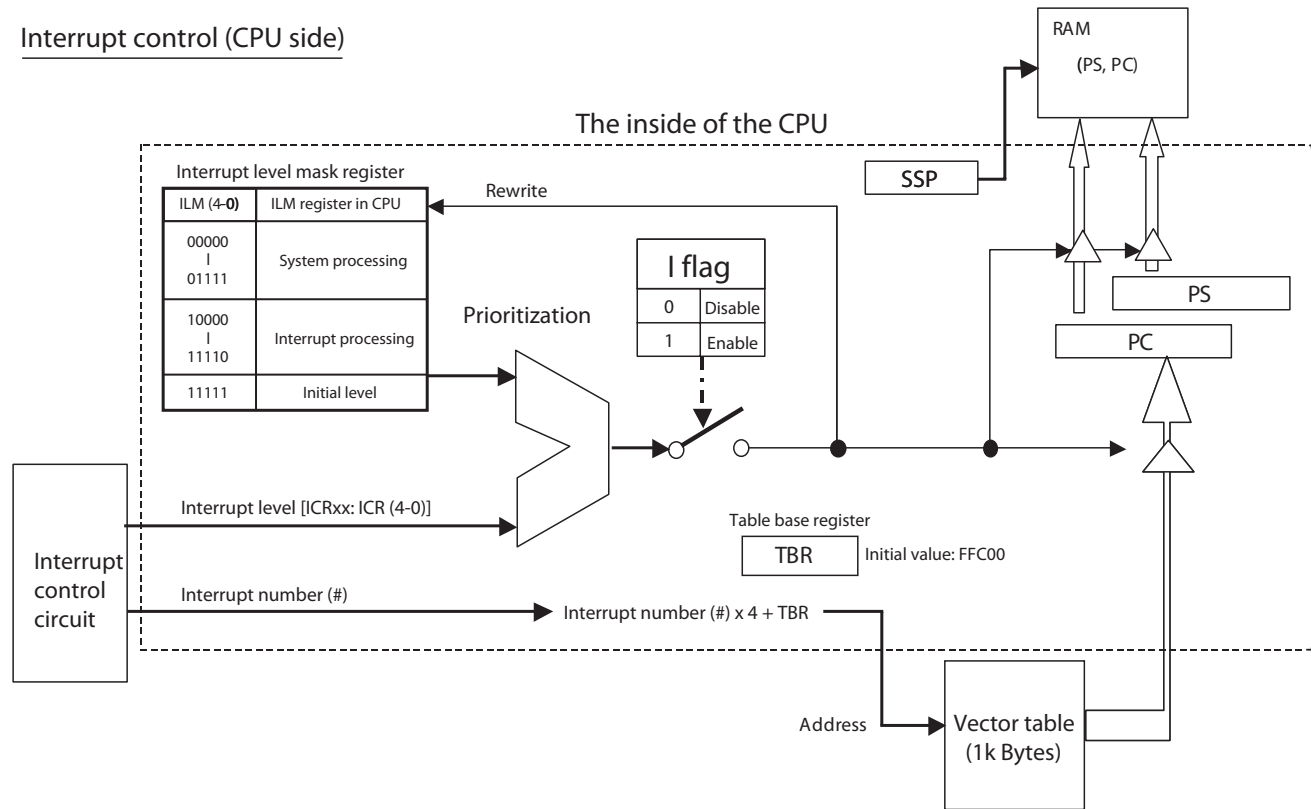


Figure 24.3-2 Configuration Diagram of Interrupt Control (CPU side)

Interrupt control (CPU side)



MB91460 Series

24.4. Registers

24.4.1 ICR00 - ICR63: Interrupt Control Register

The 64 ICR registers specify the interrupt level of each interrupt request.

ICR00	#16	External Interrupt 0	: Address 0440 _H	(Access: Byte)
	#17	External Interrupt 1		
...
ICR63	#142	DMA Controller	: Address 047F _H	(Access: Byte)
	#143	Main/Sub Oscillator Stability Wait		

Note: For the complete ICR assignments, please [See "Interrupt Vector Table" on P. 113](#)

ICRxx (Interrupt Control Registers) are the registers in the interrupt controller which specify the interrupt level for each interrupt request. Each ICR handles 2 interrupt request inputs: a “lower” and a “upper” request.

For example, ICR00 handles the external interrupts 0 and 1:

ICR00	#16	External Interrupt 0	: Address 0440 _H	(Access: Byte)
	#17	External Interrupt 1		

External Interrupt 0 is called the “lower” interrupt request, whereas External Interrupt 1 is the “upper” one.

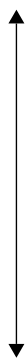
These 2 interrupt sources can be disabled independently by control bits IDBL0, IDBL1 in the ICR.

- **ICR00 – ICR63**

7	6	5	4	3	2	1	0	bit
–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	
–	–	–	1	1	1	1	1	Initial value
RX/WX	RX/WX	RX/WX	R/WX	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- Bit 7-5: Undefined. Writing does not affect the operation. The read value is indeterminate.
- Bit 4-0: Interrupt level setting bits

ICR4-ICR0 bits	Interrupt level	Description
0000-01110	0-14	Reserved for system (cannot be set by user)
01111	15	NMI
10000	16	<div style="text-align: center;"> The highest level (High)  (Low) The lowest level </div>
10001	17	
10010	18	
10011	19	
10100	20	
10101	21	
10110	22	
10111	23	
11000	24	
11001	25	
11010	26	
11011	27	
11100	28	
11101	29	
11110	30	
11111	31	Disable interrupts

- The interrupt level setting bit specifies the interrupt level of the corresponding interrupt request.
- When the interrupt level set to the interrupt control register is the same as, or higher than the level mask value set to the ILM register of the CPU (See “[ILM: Interrupt Level Mask Register](#)” on P. 212), the interrupt request is masked by the CPU side.

MB91460 Series

24.4.2 HRCL (Hold Request Cancel Level register)

This register defines the level at which a hold request cancel request is generated.

- **HRCL: Address 0039 (Access: Byte)**

7	6	5	4	3	2	1	0	bit
MHALTI	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	
0			1	1	1	1	1	Initial value (INITX pin input, watchdog reset)
0			1	1	1	1	1	Initial value (Software reset)
R/W			R	R/W	R/W	R/W	R/W	Attribute

(See "Meaning of Bit Attribute Symbols" on P. 15 for details of the attributes.)

- **Bit7: MHALTI**
The MHALTI bit indicates an NMI request. This bit is set to 1 by an NMI request. The NMI request is cleared if this bit is set to 0.
- **Bits6-5: Reserved bits.**
- **Bits 4-0: LVL4 to 0**
These bits define the interrupt level at which a hold request cancel request is generated for the bus master.
If an interrupt request with a stronger interrupt level than specified in this register is generated, a hold request cancel request is sent to the bus master. The LVL4 bit is always 1 and cannot be rewritten to 0.

24.4.3 Interrupt Vector

Interrupt vector that corresponds to a vector number (#) with TBR register set to 0FFC00h (initial value):

#00	: Address	FFFFCh
#01	: Address	FFFF8h
}	}	}
#07	: Address	FFFE0h
}	}	}
#63	: Address	FFF00h
}	}	}
#143	: Address	FFDC0h

32 bits

- Set the address of each interruption handling routine to the corresponding vector.
- The address of a vector = TBR (table vector register) + {3FCH - 4 x vector number (#)}
- EIT used by system (#0-#14) (See "EIT Interrupt Level" on P. 219)

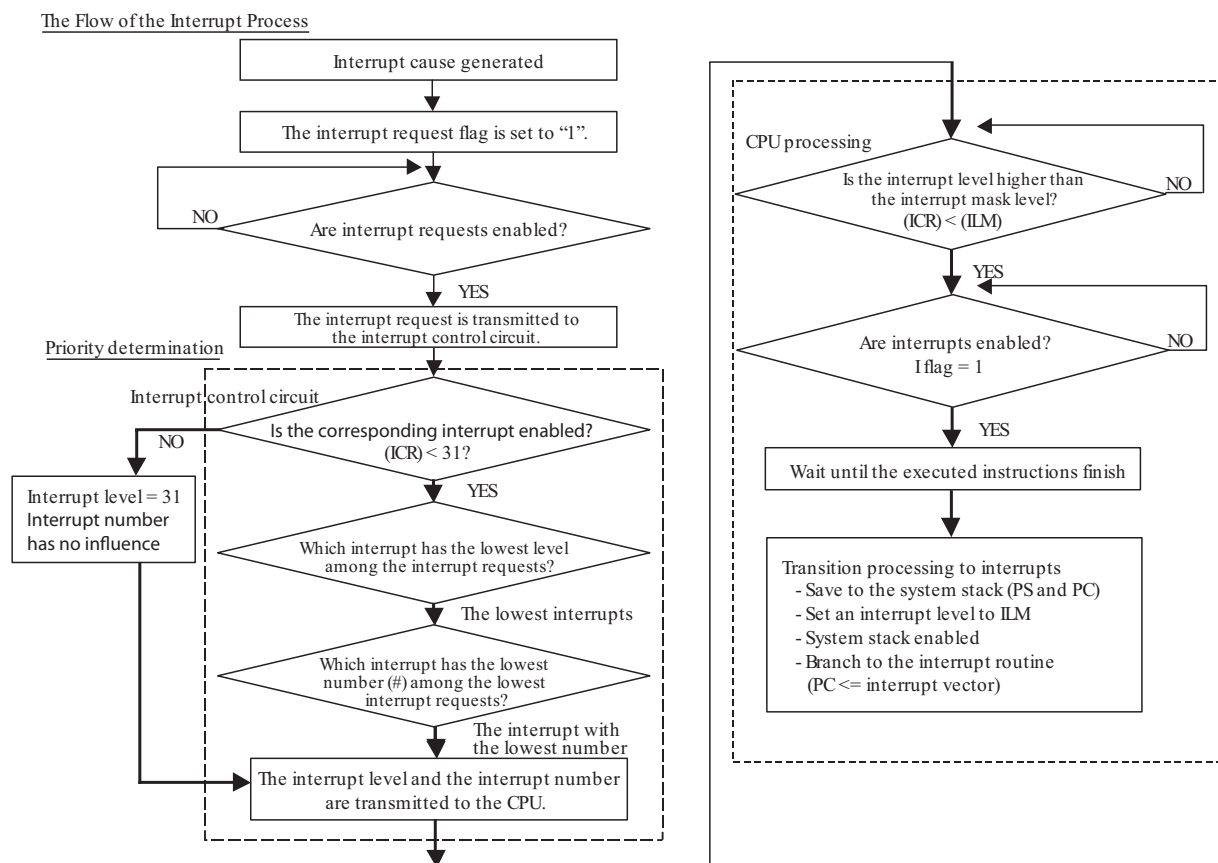
Interrupt number	Interrupt level (fixed)	interrupt event
#0	0	Reset vector
#1	1	Mode vector
#2-#4	—	Reserved for system
#5	5	CPU Supervisor Mode
#6	6	Memory Protection Exception
#7	7	Coprocessor absence trap
#8	8	Coprocessor error trap
#9	9	INTE instruction
#10	10	Instruction break exception
#11	11	Operand break trap
#12	12	Step trace trap
#13	13	MNI request (TOOL)
#14	14	Undefined-instruction exception
#15	15	NMI request

Please refer also to section [0.1. Interrupt Vector Table \(Page No.31\)](#).

24.5. Operation

The following section explains priority determination operation of interrupt control.

Figure 24.5-1 Priority determination operation of interrupt control



■ Priority determination

- The interrupt control circuit selects the highest priority event from those that have been generated simultaneously, and outputs the event's interrupt level (ICR) and interrupt number (#) to the CPU.
- The priority level criteria of an interrupt cause are the following conditions.
 - The value of the interrupt level is not 31. (31 is "interrupt disable")
 - The events with the smallest interrupt level.
 - Among these, the event that has the smallest interrupt number.
- If nothing is applicable by the above-mentioned criteria, interrupt level 31 (11111_B) is sent to the CPU. In this case, the interrupt number has no influence.

■ Hold request cancel request (HRCL)

To process an interrupt with a high priority during CPU hold, request the hold request generator to cancel the request. In the HRCL register, define the interrupt level at which a cancel request is to be generated.

• Generation conditions

If an interrupt cause with a stronger interrupt level than specified in the HRCL register is generated, a hold request cancel request is generated.

Interrupt level of the HRCL register > Interrupt level after the priority evaluation -> Cancel request generated

Interrupt level of the HRCL register ≤ Interrupt level after the priority evaluation -> Cancel request not generated

Unless the interrupt cause that generated the cancel request is cleared, the cancel request remains valid and prevents any DMA transfers from occurring. Be sure to clear the corresponding interrupt cause.

While a NMI is used, the MHALTI bit of the HRCL register is set to 1. As long as this bit is not cleared, the cancel request is valid. To put the CPU in the hold status, clear the MHALTI bit.

- **Definable level**

The HRCL register can be set to 10000_B to 11111_B as with the ICR.

This bit, if set to 11111_B, generates a cancel request for all the interrupt levels. This bit, if set to 10000_B, generates a cancel request only for an NMI.

This table shows how to define the interrupt level at which a hold request cancel request is generated.

Table 24.5-1 Interrupt level at which a cancel request is generated

HRCL register	Interrupt level at which a cancel request is generated
16	NMI only
17	NMI and Interrupt Level 16
18	NMI and Interrupt Levels 16 and 17
~	~
31	NMI and Interrupt Levels 16 through 30 (initial value)

After resetting, the DMA transfer is disabled at all interrupt levels. No DMA transfer is performed even though an interrupt occurs. In this case, set the HRCL register to the required value.

■ Recovery from standby mode (STOP/SLEEP)

This circuit provides a function for recovering from STOP state if an interrupt request is generated. If at least one interrupt request from a peripheral resource (with an interrupt level other than 11111), including an NMI, is generated, a request for recovery from STOP state is sent to the clock controller.

Since the priority evaluation unit restarts when the clock is re-supplied after recovery from STOP state, the CPU executes instructions until the priority evaluation unit provides a result.

Recovery from SLEEP state is the same as described above.

The registers of this module can be accessed even during SLEEP state.

- **Precautions**

- The NMI request also initiates recovery from STOP state. However, configure the NMI so that a valid input is detected even in STOP state.
- For an interrupt cause that you do not want to initiate recovery from STOP or SLEEP state, set the interrupt level to 11111 in the control register of the corresponding peripheral resource.

■ Example of using the HRCL function

To have the CPU perform processing with a high priority during the DMA transfer, have DMA cancel the hold request, thus canceling the hold status of the CPU. This section explains how to use an interrupt to have DMA cancel the hold request, i.e., enable the CPU implementing a higher priority operation.

- **Control register**

- **HRCL (Hold Request Cancel Level setting register):**

If an interrupt with an interrupt level stronger than specified in this register is generated, a hold request cancel request is sent to DMA. This register defines the reference level for this purpose.

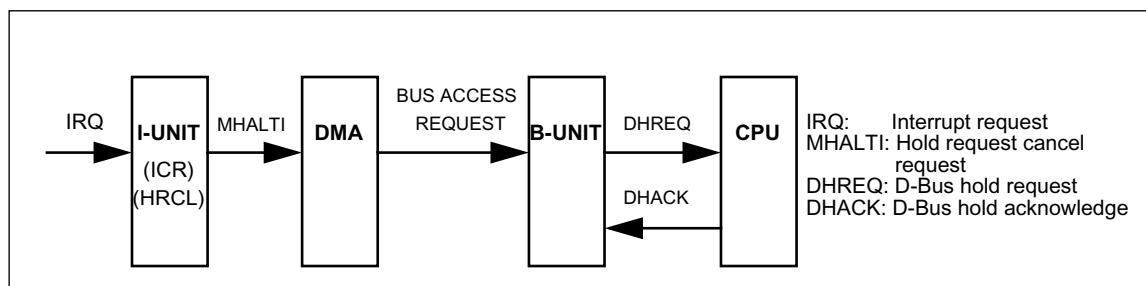
- **ICR:**

In the ICR corresponding to the interrupt cause to be used, define a level stronger than that specified in the HRCL register.

- Hardware configuration

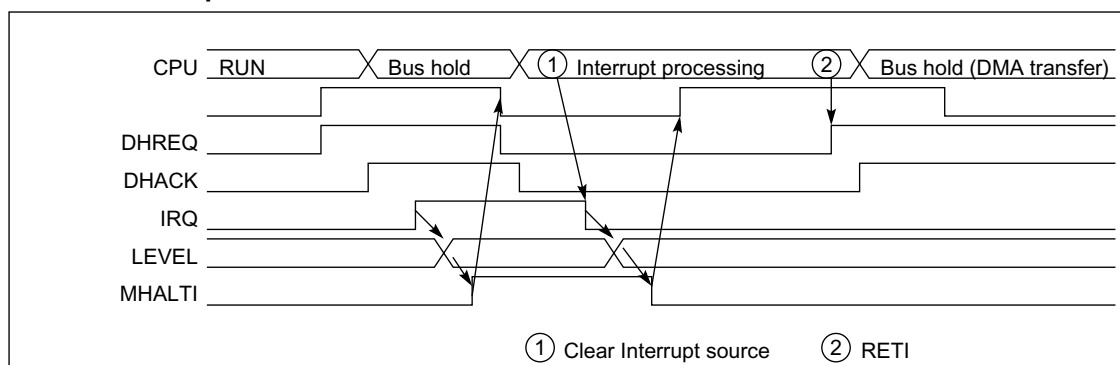
The signal flow is as follows:

Figure 24.5-2 Hardware configuration for Hold request cancel request



- Sequence

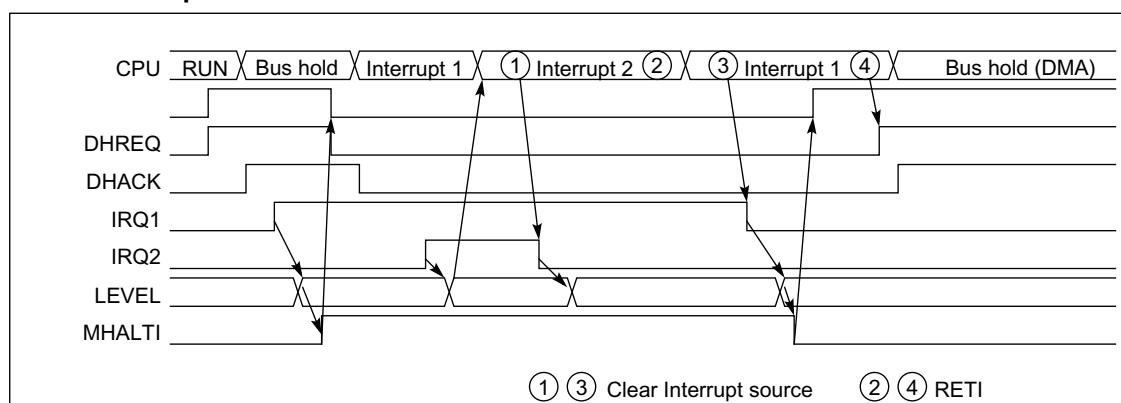
Figure 24.5-3 Interrupt level HRCL > a



If an interrupt request occurs, it changes the interrupt level. The interrupt level, if it is stronger than that specified in the HRCL register, makes MHALTI active for DMA. This will cause DMA to set the access request low and the CPU to recover from hold status to perform the interrupt processing.

The following figure shows the case of multiple interrupts:

Figure 24.5-4 Interrupt level HRCL > a > b



The above example shows that, while Interrupt Routine 1 is being executed, an interrupt with a higher priority occurs. While a higher interrupt level than that specified in the HRCL register exists, DHREQ remains low.

- Precautions

Be especially careful of the relationship between the interrupt levels defined in the HRCL register and those defined in the ICR.

24.6. Setting

Table 24.6-1 Setting Required to Use Interrupts

Setting	Setting Registers	Setting Procedure
Setting the interrupt level	Interrupt control registers (ICR00 to ICR63)	See 7.1
Clearing the interrupt request flags	See the corresponding chapter for each peripheral function.	—
Enabling interrupt requests	See the corresponding chapter for each peripheral function.	—
I flag setting	CCR register	See 7.5

*: For the setting procedure, refer to the section indicated by the number.

Table 24.6-2 Setting that Requires the Setting within Interrupt Processing

Setting	Setting Registers	Setting Procedure
Clearing the interrupt request flags	See the corresponding chapter for each peripheral function.	—

24.7. Q & A

24.7.1 How to set interrupt levels

Set by Interrupt control registers (ICR00 to ICR63).

It is necessary to set interrupt levels in advance to the control registers of the applicable interrupts.

	Interrupt control registers ICR00 to ICR63
How to configure to the highest level	Set 16.
How to configure to a level	Set any level (from 16 to 30).
How to configure to the lowest level	Set 30.
When is the interrupt not used	Set 31 (interrupt disable).

- Since the bit of the interrupt control register (ICR[4]) is fixed to “1”, 0 to 15 cannot be set to a register.

24.7.2 How to enable interrupts

To enable interrupts, all of the following three settings should be set:

- Set the value 16 to 30 to the applicable register in the interrupt control registers (ICR00-ICR63).
- Set the interrupt request enable bit of the applicable peripheral function to “1” (enable) (See the chapter for the corresponding peripheral function).
- Set the interrupt enable bit (I) to “1.”

24.7.3 How to disable interrupts

To disable interrupts, at least one of the following three settings should be set:

- Set the value 31 to the applicable register in the interrupt control registers (ICR00-ICR63).
- Set the interrupt request enable bit of the applicable peripheral function to “0” (disable).
- Set the interrupt enable bit (I) to “0” (disable all interrupts.)

MB91460 Series

24.7.4 How to set an I flag

→Using C programming language:

I flag is set to “1” (interrupt enable) by writing `__EI();` (EI() is a macro function).

I flag is set to “0” (interrupt disable) by writing `__DI();` (DI() is a macro function).



Two underscores

24.8. Caution

Interrupt flags are not cleared automatically. Make sure to clear them in the interrupt process.

(They are usually cleared by writing “0” to the interrupt flag, however, there are some exceptions depending on the type of peripheral functions. See the chapter for the corresponding peripheral function.)

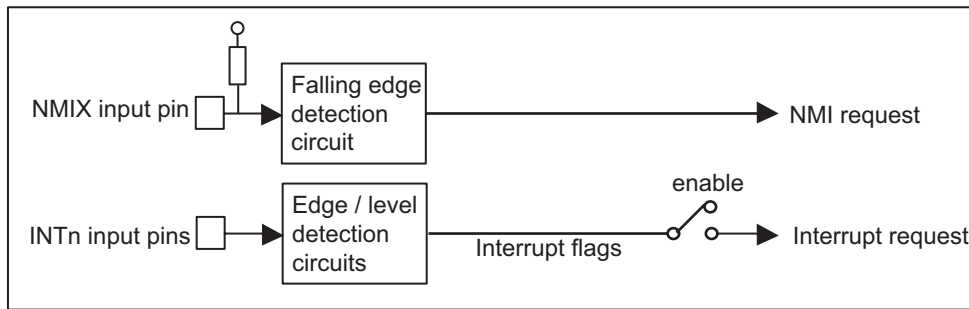
Chapter 25 External Interrupt and NMI

25.1. Overview

An external interrupt is generated whenever an external interrupt request level is applied to an external interrupt input pin INTn, and the respective interrupt request is enabled.

A non-maskable interrupt (NMI) is generated whenever a falling edge is applied to the NMIX pin.

Figure 25.1-1 Block diagram of external interrupt



Note: INTn with n = 0 to 15 or 0 to 31.

The output of the above figure (P. 441) is an input to the Interrupt control (See "Block diagram of Interrupt Control" on P. 429)

25.2. Features

- Quantity : up to 16 (INT input -- 16 channels: INT0-INT15)
- : up to 32 (INT input -- 32 channels: INT0-INT31) for MB91FV460B and MB91F469Q
- : 1 NMIX input on selected devices.

Note: To find out which resources are implemented on a device, please refer to section 2.3.3 MB91460 Series Resource Lineup (Page No.29).

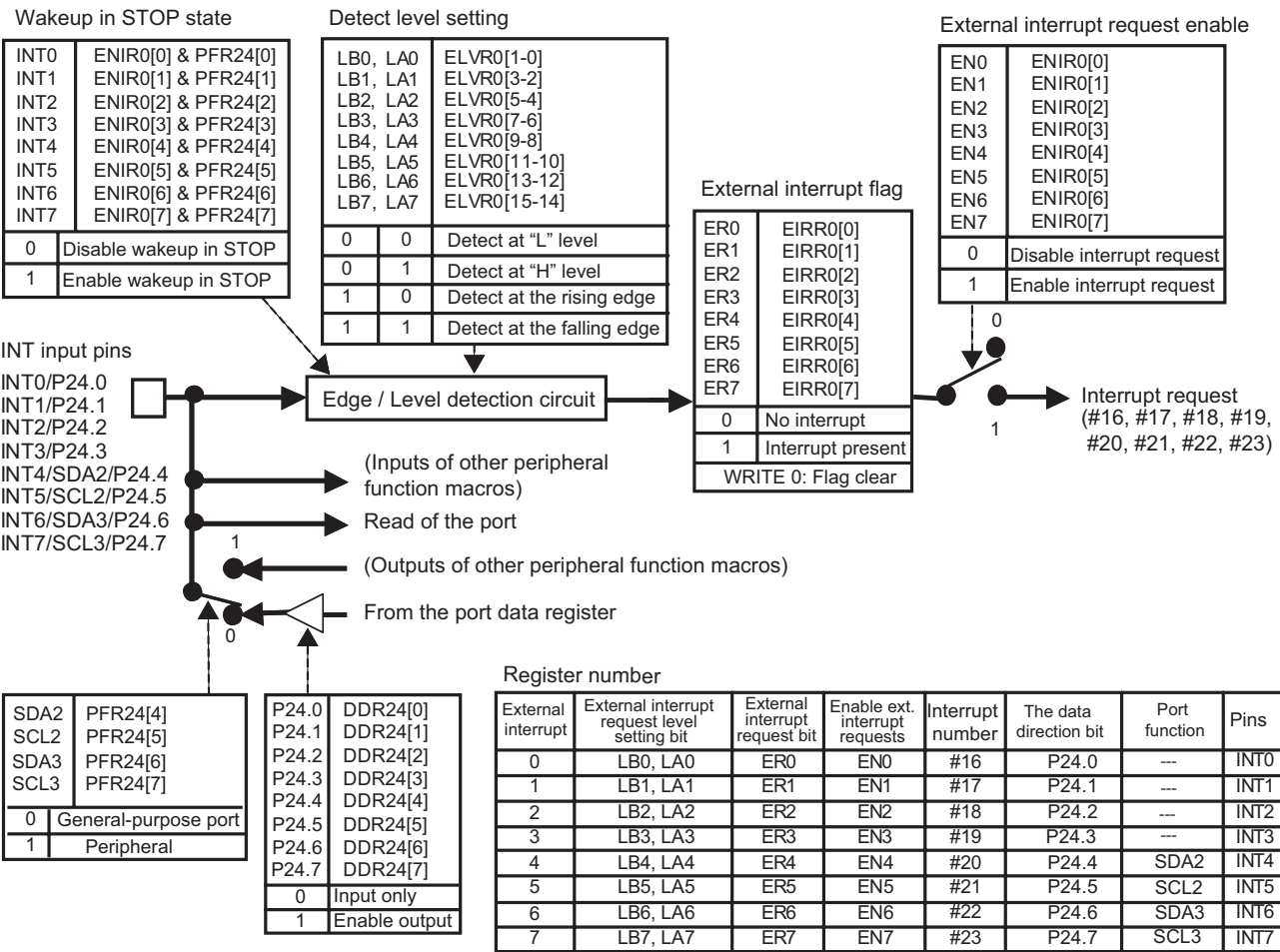
- External interrupt request level: 4 levels
 - "L" level
 - "H" level
 - Rising edge
 - Falling edge
- Non-maskable interrupt request: 1 level
 - Falling edge

25.3. Configuration

25.3.1 Configuration External Interrupts 0 to 7

Figure 25.3-1 Configuration Diagram for external interrupts 0-7

External interrupts 0 - 7

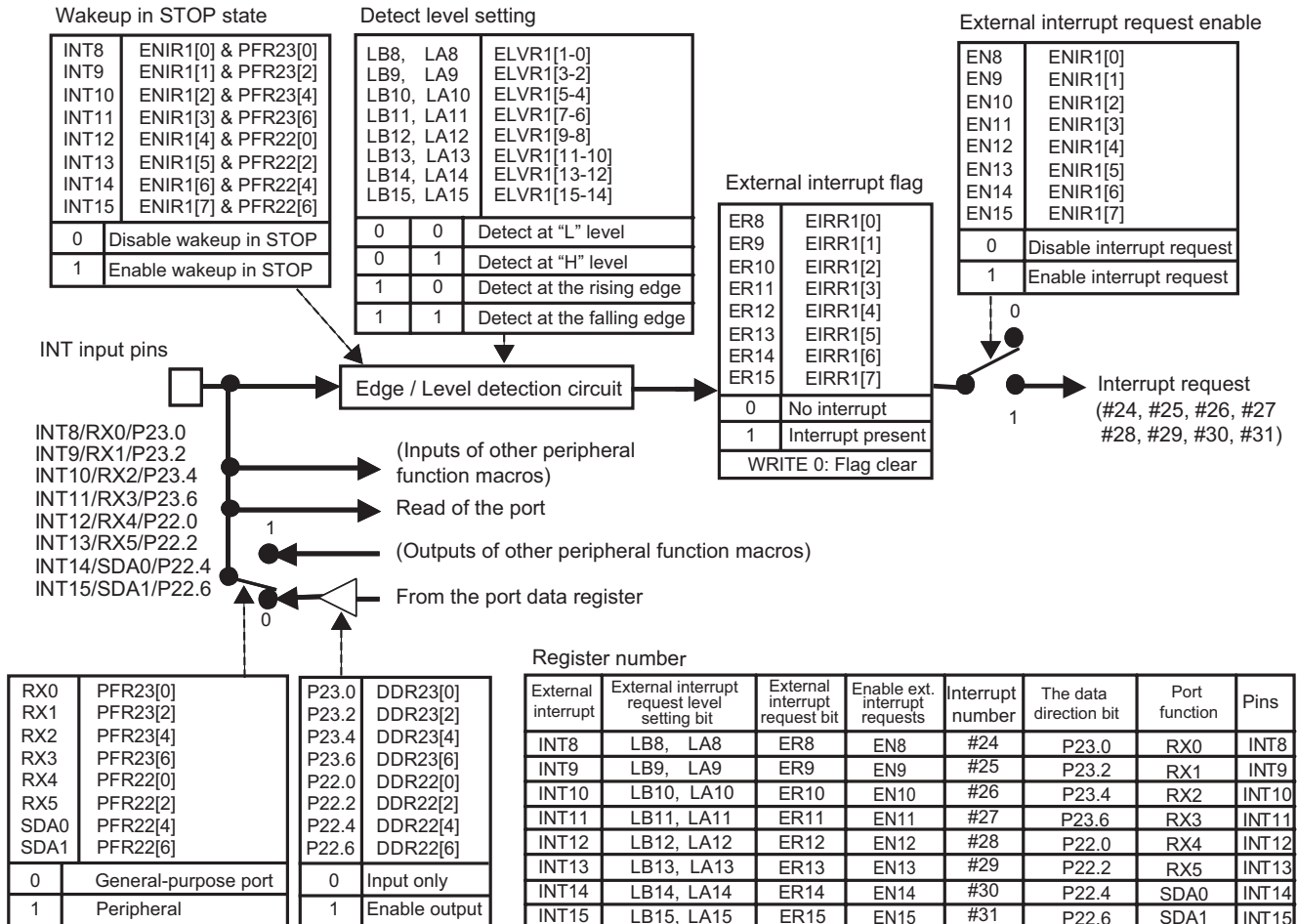


MB91460 Series

25.3.2 Configuration External Interrupts 8 to 15

Figure 25.3-2 Configuration Diagram for external interrupts 8-15

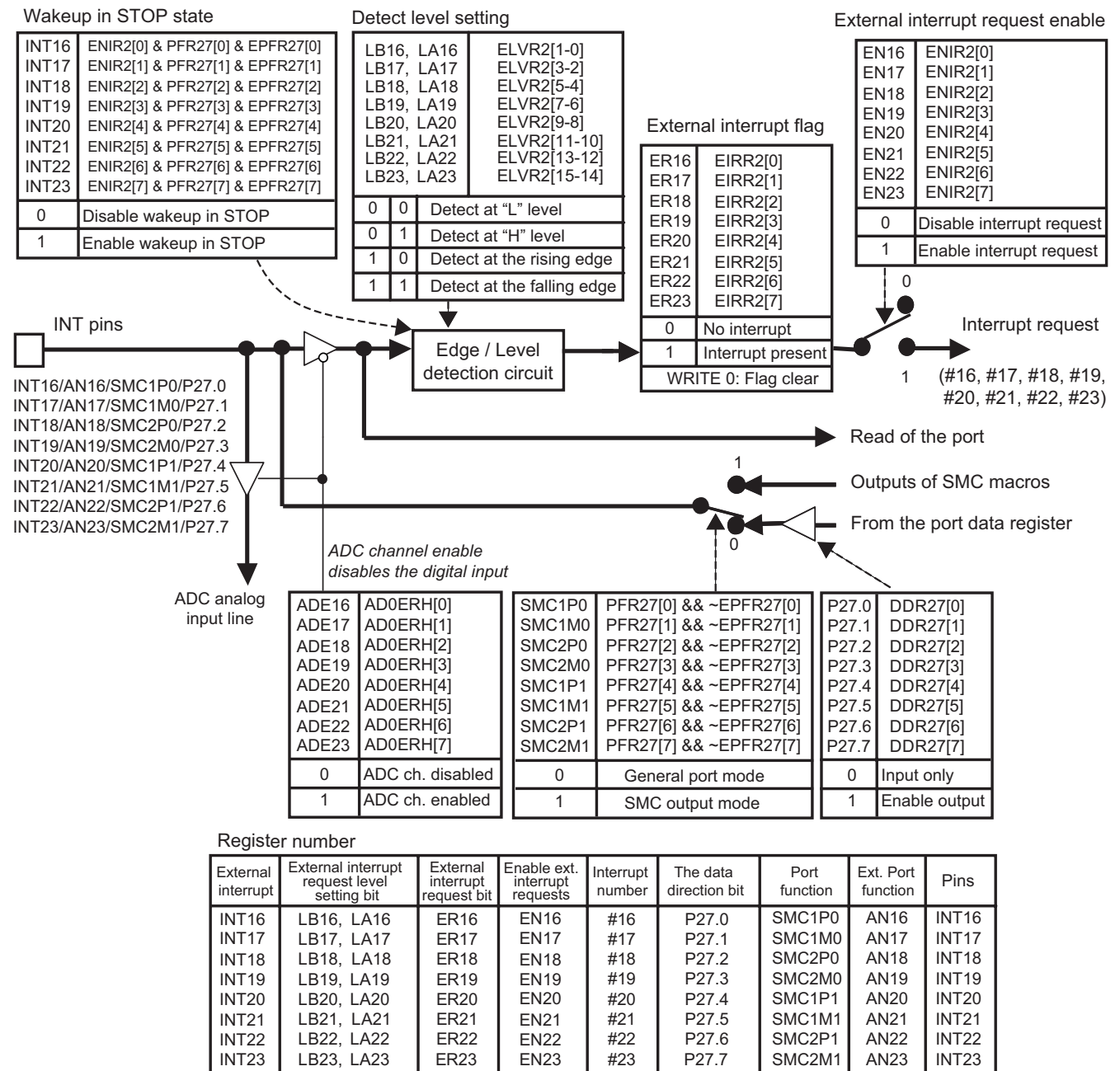
External interrupts 8 - 15



25.3.3 Configuration External Interrupts 16 to 23

Figure 25.3-3 Configuration Diagram for external interrupts 16-23

External interrupts 16 - 23

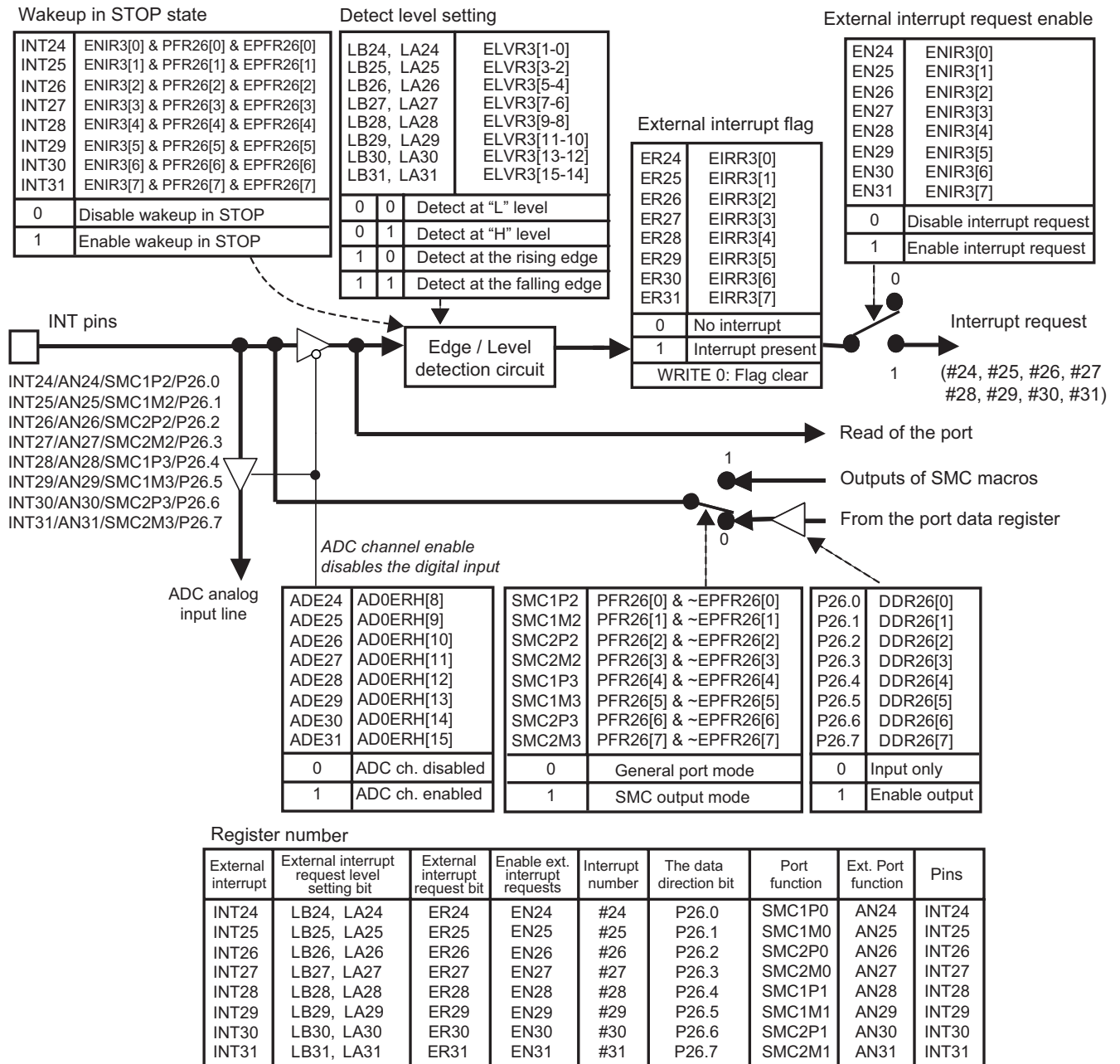


MB91460 Series

25.3.4 Configuration External Interrupts 24 to 31

Figure 25.3-4 Configuration Diagram for external interrupts 24-31

External interrupts 24 - 31



25.3.5 Register List External Interrupts 0 to 7

Figure 25.3-5 Register List External Interrupts 0-7

External interrupt (0-7)																		
Address	Bit	7	6	5	4	3	2	1	0									
00030H		ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EIRR0	(External interrupt cause 0)							
00031H		EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	ENIR0	(External interrupt enable 0)							
00032H	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR0
(External interrupt request level 0)																		
00D58H	Bit	7	6	5	4	3	2	1	0									
		P24.7	P24.6	P24.5	P24.4	P24.3	P24.2	P24.1	P24.0	DDR24	(Data direction)							
00D98H		P24.7	P24.6	P24.5	P24.4	P24.3	P24.2	P24.1	P24.0	PFR24	(Port function)							
00440H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR00	(Interrupt level #16/#17)							
00441H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01	(Interrupt level #18/#19)							
00442H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02	(Interrupt level #20/#21)							
00443H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03	(Interrupt level #22/#23)							
0FFFBCH		32Bits														(Interrupt vector #16)		
0FFFB8H		32Bits														(Interrupt vector #17)		
0FFFB4H		32Bits														(Interrupt vector #18)		
0FFFB0H		32Bits														(Interrupt vector #19)		
0FFFA8H		32Bits														(Interrupt vector #20)		
0FFFA4H		32Bits														(Interrupt vector #21)		
0FFFA0H		32Bits														(Interrupt vector #22)		
0FFFA0H		32Bits														(Interrupt vector #23)		

* See the chapter of "Interrupt Control" about ICR register and interrupt vectors.

MB91460 Series

25.3.6 Register List External Interrupts 8 to 15

Figure 25.3-6 Register List External Interrupts 8-15

External interrupt (8-15)

Address	Bit	7	6	5	4	3	2	1	0									
00034H		ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	EIRR1	(External interrupt cause 1)							
00035H		EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	ENIR1	(External interrupt enable 1)							
Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
00036H		LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	ELVR1
																		(External Interrupt request level 1)
Address	Bit	7	6	5	4	3	2	1	0									
00D56H		---	P22.6	---	P22.4	---	P22.2	---	P22.0	DDR22	(Data direction)							
00D96H		---	P22.6	---	P22.4	---	P22.2	---	P22.0	PFR22	(Port function)							
00D57H		---	P23.6	---	P23.4	---	P23.2	---	P23.0	DDR23	(Data direction)							
00D97H		---	P23.6	---	P23.4	---	P23.2	---	P23.0	PFR23	(Port function)							
00444H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04	(Interrupt level #24/#25)							
00445H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05	(Interrupt level #26/#27)							
00446H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06	(Interrupt level #28/#29)							
00447H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07	(Interrupt level #30/#31)							
Address												32Bits						(Interrupt vector #24)
0FFF9CH												32Bits						(Interrupt vector #24)
0FFF98H												32Bits						(Interrupt vector #25)
0FFF94H												32Bits						(Interrupt vector #26)
0FFF90H												32Bits						(Interrupt vector #27)
0FFF8CH												32Bits						(Interrupt vector #28)
0FFF88H												32Bits						(Interrupt vector #29)
0FFF84H												32Bits						(Interrupt vector #30)
0FFF80H												32Bits						(Interrupt vector #31)

* See the chapter of "Interrupt Control" about ICR register and interrupt vectors.

25.3.7 Register List External Interrupts 16 to 23

Figure 25.3-7 Register List External Interrupts 16-23

External interrupt (16-23)

Address	Bit	7	6	5	4	3	2	1	0										
00C04H		ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16	EIRR2	(External interrupt cause 2)								
00C05H		EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16	ENIR2	(External interrupt enable 2)								
00C06H	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
		LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16	ELVR2	
		(External interrupt request level 2)																	
00D5BH	Bit	7	6	5	4	3	2	1	0										
		P27.7	P27.6	P27.5	P27.4	P27.3	P27.2	P27.1	P27.0	DDR27	(Data direction)								
00D9BH		P27.7	P27.6	P27.5	P27.4	P27.3	P27.2	P27.1	P27.0	PFR27	(Port function)								
00DDBH		P27.7	P27.6	P27.5	P27.4	P27.3	P27.2	P27.1	P27.0	EPFR27	(Extended Port function)								
001A0H		ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16	AD0ERH	(ADC0 channel enable)								
00440H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR00	(Interrupt level #16/#17)								
00441H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01	(Interrupt level #18/#19)								
00442H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02	(Interrupt level #20/#21)								
00443H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03	(Interrupt level #22/#23)								
0FFFBCH		32Bits																	(Interrupt vector #16)
0FFFB8H		32Bits																	(Interrupt vector #17)
0FFFB4H		32Bits																	(Interrupt vector #18)
0FFFB0H		32Bits																	(Interrupt vector #19)
0FFFA8H		32Bits																	(Interrupt vector #20)
0FFFA4H		32Bits																	(Interrupt vector #21)
0FFFA0H		32Bits																	(Interrupt vector #22)
0FFFA0H		32Bits																	(Interrupt vector #23)

* See the chapter of "Interrupt Control" about ICR register and interrupt vectors.

Note: Interrupts 16 to 31 are only available on MB91FV460B and MB91F469Q.

MB91460 Series**25.3.8 Register List External Interrupts 24 to 31****Figure 25.3-8 Register List External Interrupts 24-31**

External interrupt (24-31)

Address	Bit	7	6	5	4	3	2	1	0									
00C08H		ER31	ER30	ER29	ER28	ER27	ER26	ER25	ER24	EIRR3	(External interrupt cause 3)							
00C09H		EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	ENIR3	(External interrupt enable 3)							
00C0AH	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28	LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24	ELVR3
		(External interrupt request level 3)																
00D5AH	Bit	7	6	5	4	3	2	1	0									
		P26.7	P26.6	P26.5	P26.4	P26.3	P26.2	P26.1	P26.0	DDR26	(Data direction)							
00D9AH		P26.7	P26.6	P26.5	P26.4	P26.3	P26.2	P26.1	P26.0	PFR26	(Port function)							
00DDAH		P26.7	P26.6	P26.5	P26.4	P26.3	P26.2	P26.1	P26.0	EPFR26	(Extended Port function)							
001A0H		ADE31	ADE30	ADE29	ADE28	ADE27	ADE26	ADE25	ADE24	AD0ERH	(ADC0 channel enable)							
00444H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04	(Interrupt level #24/#25)							
00445H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05	(Interrupt level #26/#27)							
00446H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06	(Interrupt level #28/#29)							
00447H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07	(Interrupt level #30/#31)							
Address																		
0FFF9CH		32Bits															(Interrupt vector #24)	
0FFF98H		32Bits															(Interrupt vector #25)	
0FFF94H		32Bits															(Interrupt vector #26)	
0FFF90H		32Bits															(Interrupt vector #27)	
0FFF8CH		32Bits															(Interrupt vector #28)	
0FFF88H		32Bits															(Interrupt vector #29)	
0FFF84H		32Bits															(Interrupt vector #30)	
0FFF80H		32Bits															(Interrupt vector #31)	

* See the chapter of "Interrupt Control" about ICR register and interrupt vectors.

Note: Interrupts 16 to 31 are only available on MB91FV460B and MB91F469Q.

Note: See [“Chapter 24 Interrupt Control \(Page No.429\)”](#) about ICR register and interrupt vectors.

25.4. Control/Status Register Description

25.4.1 Control /Status Registers ENIR, ELVR, EIRR

The external interrupt control and status registers are located at the following addresses.

Address	Register	Interrupt Channels	Register
000030 _H	EIRR0	INT 0-7	External interrupt request register
000031 _H	ENIR0		External interrupt enable register
000032 _H	ELVR0		External interrupt level register (16-bit)
000034 _H	EIRR1	INT 8-15	External interrupt request register
000035 _H	ENIR1		External interrupt enable register
000036 _H	ELVR1		External interrupt level register (16-bit)
000C04 _H	EIRR2	INT 16-23 ¹	External interrupt request register
000C05 _H	ENIR2		External interrupt enable register
000C06 _H	ELVR2		External interrupt level register (16-bit)
000C08 _H	EIRR3	INT 24-31 ¹	External interrupt request register
000C09 _H	ENIR3		External interrupt enable register
000C0A _H	ELVR3		External interrupt level register (16-bit)

1. Channels INT16 to INT31 are available on MB91FV460B and MB91F469Q only.

MB91460 Series

25.4.2 ELVR: Interrupt Request Level Register

The register selects the interrupt request level of the INT input pins (rising edge, falling edge, high level or low level).

- **ELVR0 (INT0-INT7): Address 032H (access: Half-word, Word)**

15	14	13	12	11	10	9	8	Bit
LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	Bit
LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **ELVR1 (INT8-INT15): Address 036H (access: Half-word, Word)**

15	14	13	12	11	10	9	8	Bit
LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	Bit
LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- **ELVR2 (INT16-INT23): Address C06H (access: Half-word, Word)**

15	14	13	12	11	10	9	8	Bit
LB23	LB23	LB22	LA22	LB21	LA21	LB20	LA20	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	Bit
LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **ELVR3 (INT24-INT31): Address C0AH (access: Half-word, Word)**

15	14	13	12	11	10	9	8	Bit
LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

Interrupt request level bits (LBn, LAn) are registers that configures the Edge / Level detection circuit of external interrupt.

2 bits (LBn, LAn) are assigned to each external interrupt INTn.

LBn	LAn	Description
0	0	Detection of “L” level and generation of an interrupt request.
0	1	Detection of “H” level and generation of an interrupt request.
1	0	Detection of the rising edge and generation of an interrupt request.
1	1	Detection of the falling edge and generation of an interrupt request.

When the interrupt request level is “H” level or “L” level (LAn, LBn = “00” or “01”), and when the INTn pin input is the valid level, the corresponding flag (ERn) will be re-set to “1” even if the external interrupt flag (ERn) is set to “0” by software simultaneously.

Note: n = 0 to 31

MB91460 Series

25.4.3 EIRR: External Interrupt Request Register

Status flag of an external interrupt.

- **EIRR0 (INT0-INT7): Address 030H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	
0	0	0	0	0	0	0	0	Initial value
R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **EIRR1 (INT8-INT15): Address 034H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8	
0	0	0	0	0	0	0	0	Initial value
R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **EIRR2 (INT16-INT23): Address C04H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16	
0	0	0	0	0	0	0	0	Initial value
R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **EIRR3 (INT24-INT31): Address C08H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
ER31	ER30	ER29	ER28	ER27	ER26	ER25	ER24	
0	0	0	0	0	0	0	0	Initial value
R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	R (RM1), W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

An external interrupt flag (ERn) indicates the corresponding external interrupt.

ERn	Description	
	Read value	Write value
0	No external interrupt present	Clear external interrupt flag
1	External interrupt present	No effect on operation

Note: n = 0 to 31

25.4.4 ENIR: Enable Interrupt Request Register

Enable bit of external interrupt requests.

- **ENIR0 (INT0-INT7): Address 031H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **ENIR1 (INT8-INT15): Address 035H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **ENIR2 (INT16-INT23): Address C05H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- **ENIR3 (INT24-INT31): Address C09H (access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	Bit
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

An external interrupt request enable bit (ENn) enables the corresponding external interrupt.

ENn	Description
0	External interrupt request output disable
1	External interrupt request output enable

Note: n = 0 to 31

MB91460 Series**25.4.5 I/O Port Assignments and Wakeup from STOP**

The following table lists the interrupt channel port assignments and how to enable the wakeup from STOP mode. Note that the ENIR register bits must be set additionally per channel.

INT channel	IO Port	Registers to be set			Comments
		ENIR	PFR	EPFR	
INT 0 to INT 7	GP24_0 to GP24_7	ENIR0_0=1 to ENIR0_7=1	PFR24_0=1 to PFR24_7=1	-	Wakeup from STOP mode is enabled by PFR=1
INT 8	GP23_0	ENIR1_0=1	PFR23_0=1	-	CAN 0 Rx, wakeup is enabled by PFR=1
INT 9	GP23_2	ENIR1_1=1	PFR23_2=1	-	CAN 1 Rx, wakeup is enabled by PFR=1
INT 10	GP23_4	ENIR1_2=1	PFR23_4=1	-	CAN 2 Rx, wakeup is enabled by PFR=1
INT 11	GP23_6	ENIR1_3=1	PFR23_6=1	-	CAN 3 Rx, wakeup is enabled by PFR=1
INT 12	GP22_0	ENIR1_4=1	PFR22_0=1	-	CAN 4 Rx, wakeup is enabled by PFR=1
INT 13	GP22_2	ENIR1_5=1	PFR22_2=1	-	CAN 5 Rx, wakeup is enabled by PFR=1
INT 14	GP22_4	ENIR1_6=1	PFR22_4=1	-	I ² C 0 SDA, wakeup is enabled by PFR=1
INT 15	GP22_6	ENIR1_7=1	PFR22_6=1	-	I ² C 1 SDA, wakeup is enabled by PFR=1
INT 16 to INT 23	GP27_0 to GP27_7	ENIR2_0=1 to ENIR2_7=1	PFR27_0=1 to PFR27_7=1	EPFR27_0=1 to EPFR27_7=1	Wakeup in STOP is enabled by setting PFR27[p]=1, EPFR27[p]=1, ENIR2[i-16]=1 and ADERH[an-16]=0. *1
INT 24 to INT 31	GP26_0 to GP26_7	ENIR3_0=1 to ENIR3_7=1	PFR26_0=1 to PFR26_7=1	EPFR26_0=1 to EPFR26_7=1	Wakeup in STOP is enabled by setting PFR26[p]=1, EPFR26[p]=1, ENIR3[i-24]=1 and ADERH[an-16]=0. *2

1. p = 0 to 7 (IO bit number), i = 16 to 23 (interrupt number), an = 16 to 23 (adc channel number),
ENIR2 = Enable Interrupt register,
ADERH = ADC Channel enable register in ADC0 (for channels 16 to 31)
2. p = 0 to 7 (IO bit number), i = 24 to 31 (interrupt number), an = 24 to 31 (adc channel number),
ENIR3 = Enable Interrupt register,
ADERH = ADC Channel enable register in ADC0 (for channels 16 to 31)

Note that, if the ADC channel is enabled (ADERH[an]=1), all digital input functionality on the pin is disabled: CMOS-Schmitt, Automotive and CMOS-2 level input lines keep their value (bus holder behaviour), while the TTL input line is tied to low level. In this state, external interrupts cannot be used.

25.4.6 Interrupt Re-location

On some devices, the external interrupts can be relocated to other pins. The relocation can be enabled by control registers (for example, on MB91460P or MB91460T series), or the relocation is fixed (MB91460B, MB91460M, MB91460Q series).

MB91FV460B can emulate the interrupt relocation by applying a “series mode”. For a description of the Series Modes please refer to section [3.8.1 MB91460 Series Modes \(Page No.143\)](#).

MB91460 Series	INT channel	re-located IO Port	Comments
MB91460P/ MB91460T	INT0 to INT7	GP00_0 to GP00_7	Re-location enabled by setting bit PPMUX.PR0. For wakeup, ENIR and PFR24 must be set.
MB91460B	INT4 to INT7	GP05_0 to GP05_3	Re-location is fixed in “B” mode. For wakeup, ENIR and PFR24 must be set.
	INT0	GP08_1	In internal vector fetch mode (MD=000) and with external bus disabled only. For wakeup, PFR24_0 must be set.
MB91460M	INT14-M INT15-M	GP22_1 GP22_3	Re-location is fixed in “M” mode. For wakeup, ENIR and EPFR22_1/3 must be set.
MB91460Q	INT4-Q INT5-Q INT10-Q INT11-Q INT12-Q	GP34_4 GP34_5 GP34_0 GP35_0 GP35_4	Re-location is fixed in “Q” mode. On ports 34/35, PFR=1 and EPFR=0 will enable LCD function. In this case, the digital input lines are disabled, no wakeup is possible. For wakeup, ENIR must be set and the ports must be set to input direction (DDR34/35=0).
	INT14-Q	GP24_4	For wakeup, ENIR must be set and the port must be set to input direction (DDR24_4=0).

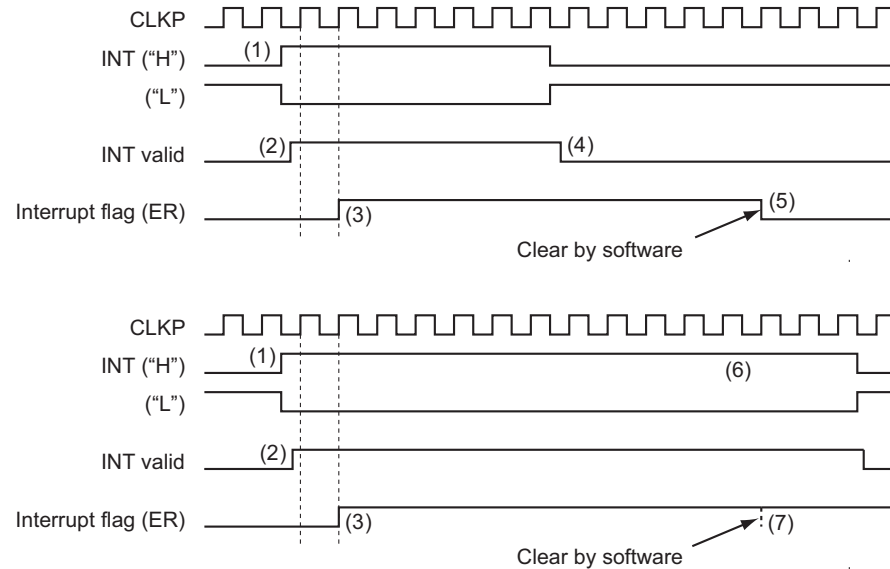
MB91460 Series

25.5. Operation

25.5.1 Level and Edge Detection

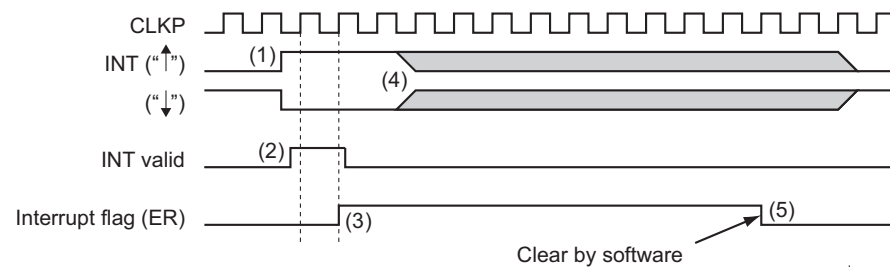
Figure 25.5-1 External Interrupt: Level and Edge Detection

Level detection:



- (1) A valid level (either "H" or "L" depending on ELVR setting) is applied on the interrupt pin.
- (2) After a noise filter, the valid input level is detected and stored internally.
- (3) The interrupt flag (ER) is set after 2 rising edges of CLKP.
- (4) The external signal has disappeared (cleared by the interrupt service routine).
- (5) The CPU clears the interrupt flag.
- ... or ...
- (6) The external signal has not yet disappeared.
- (7) The CPU cannot clear the flag because the level is still valid. If the interrupt is still enabled (EN=1), the interrupt service routine is entered again.

Edge detection:



- (1) A valid edge (either rising or falling, depending on ELVR) is applied on the interrupt pin.
- (2) After a noise filter, the valid input level is detected and stored internally.
- (3) The interrupt flag (ER) is set after 2 rising edges of CLKP.
- (4) The opposite input edge can appear anytime.
- (5) The CPU clears the interrupt flag.

Note: On the INT input signal, a minimum pulse width ($> 2 \times \text{CLKP}$) is needed.

Note: On the NMIX input signal, a minimum pulse width ($> 1 \times \text{CLKP}$) is needed.

NMIX always becomes active on falling edge.

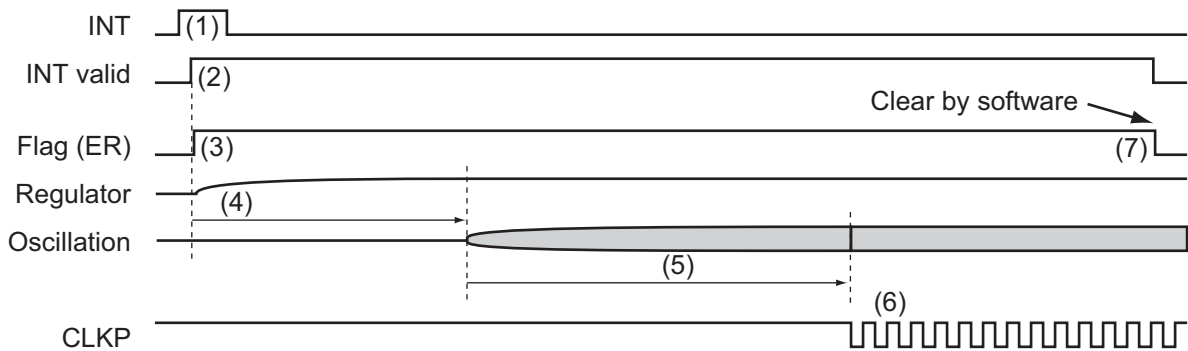
25.5.2 Wakeup from STOP State

In STOP state, the clock for the external interrupt logic (CLKP) is disabled. Therefore the interrupt request as well as NMI, are handled asynchronously.

To enable wakeup from STOP state with INT, the port function registers must be set according to figures 25.3-1, 25.3-2, 25.3-3 and 25.3-4. NMI can always wakeup the device from STOP.

Figure 25.5-2 External Interrupt: Wakeup from STOP State

Wakeup from STOP state:



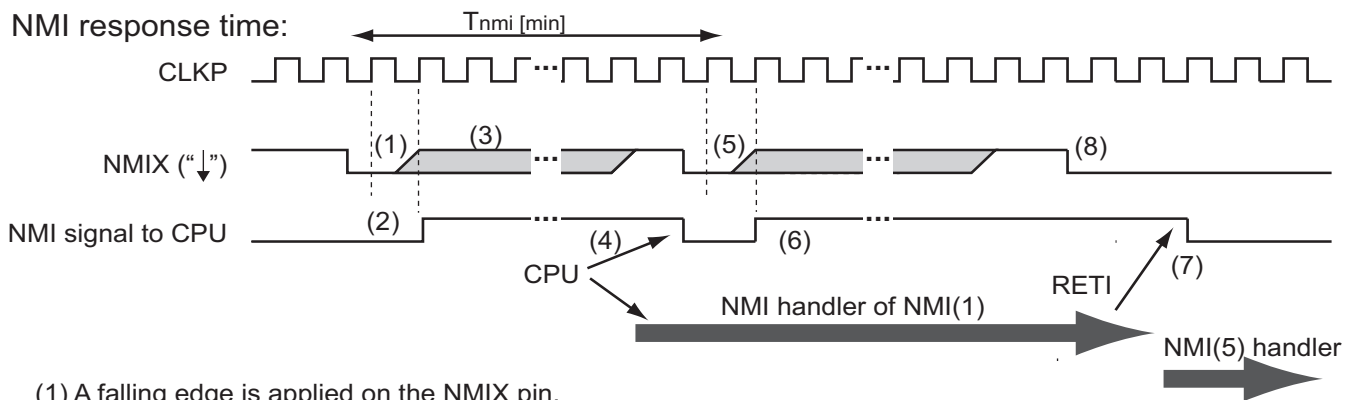
- (1) A valid edge or level (depending on ELVR setting) is applied on the interrupt pin.
- (2) After a noise filter, the valid input level/edge is detected and stored internally.
- (3) The interrupt flag (ER) is set asynchronously.
- (4) The regulator is switched to main mode, followed by regulator stabilization wait time (80 μ s typ.)
- (5) The oscillation stabilization wait time, defined by STCR.OS[1:0], is executed.
- (6) After the clocks started, the software continues running and will call the interrupt service routine.
- (7) The CPU clears the interrupt flag.

Note: On the INT and NMIX input signal in STOP mode, a minimum pulse width (> 50ns) is needed.

MB91460 Series**25.5.3 NMI Response Time**

The following figure shows the timing of two consecutive NMI events.

Figure 25.5-3 Timing of consecutive NMI events



- (1) A falling edge is applied on the NMIX pin.
- (2) The NMI signal to the CPU is generated after edge detection.
- (3) The NMIX line can switch back to high level already after 1 CLKP cycle.
- (4) The CPU has received the NMI request, clears the request by hardware and starts the NMI service handler.
- (5) The next falling edge at NMIX is sampled.
- (6) The NMI request stays active until the interrupt handler of NMI(1) is done.
- (7) After the RETI of NMI(1) interrupt handler, the handler starts again to serve NMI(5).
- (8) This falling edge on NMIX has no effect because the NMI request is already active.

The minimum time between 2 consecutive NMI events is as following:

$$T_{nmi} [min] = 4 * CLKP + 5 * CLKB$$

25.6. Setting

Table 25.6-1 Setting Required in Order to Use External Interrupts

Setting	Setting Registers	Setting Procedures *
Setting of external interrupt request level	External interrupt request level setting register (ELVR0 - ELVR3)	See 25.7.1
Set INT pin as the input.	Data direction register (DDR22, DDR23, DDR24, DDR26, DDR27) Port function register (PFR22, PFR23, PFR24, PFR26, PFR27, EPFR26, EPFR27)	See 25.7.2
Setting the interrupt levels	ICR00 to ICR07	See 25.7.3
Enable, disable and clear interrupt requests	ENIR0, ENIR1, ENIR2, ENIR3, EIRR0, EIRR1, EIRR2, EIRR3	See 25.7.5
External interrupt	External inputs →Inputs the signal to INT0 - INT31 pins.	—

Note: * For the setting procedure, refer to the section indicated by the number.

25.7. Q & A

25.7.1 What are the types and setting procedures of external interrupt request levels?

There are 4 types of external interrupt request levels: “L” level, “H” level, rising edge, and falling edge.

Carry out in Detection level bits

- ELVR0. LBx, LAx (x = 0-7)
- ELVR1. LBx, LAx (x = 8-15)
- ELVR2. LBx, LAx (x = 16-23)
- ELVR3. LBx, LAx (x = 24-31)

Detection level bit (LBn, LAn) n = 0-31	External interrupt request levels
Sets to “00”	“L” level
Sets to “01”	“H” level
Sets to “10”	rising edge
Sets to “11”	falling edge

25.7.2 How to set INT pin as the input

- Use data direction registers (DDR22, DDR23, DDR24, DDR26, DDR27).
- Use port function register (PFR22, PFR23, PFR24, PFR26, PFR27).

Table 25.7-1 External Interrupts Port Function Register Setting

To use INT...	INT Enable		Data Direction		Port Function		Other Registers	
	bits	Set to	bits	Set to	bits	Set to	bit	Set to
INT0 pin input	ENIR0.0	“1”	DDR24.0	“0”	PFR24.0	“1”	—	—
INT1 pin input	ENIR0.1	“1”	DDR24.1	“0”	PFR24.1	“1”	—	—
INT2 pin input	ENIR0.2	“1”	DDR24.2	“0”	PFR24.2	“1”	—	—
INT3 pin input	ENIR0.3	“1”	DDR24.3	“0”	PFR24.3	“1”	—	—
INT4 pin input	ENIR0.4	“1”	DDR24.4	“0”	PFR24.4	“1”	—	—
INT5 pin input	ENIR0.5	“1”	DDR24.5	“0”	PFR24.5	“1”	—	—
INT6 pin input	ENIR0.6	“1”	DDR24.6	“0”	PFR24.6	“1”	—	—

MB91460 Series

To use INT...	INT Enable		Data Direction		Port Function		Other Registers	
	bits	Set to	bits	Set to	bits	Set to	bit	Set to
INT7 pin input	ENIR0.7	"1"	DDR24.7	"0"	PFR24.7	"1"	—	—
INT8 pin input	ENIR1.0	"1"	DDR23.0	"0"	PFR23.0	"1"	—	—
INT9 pin input	ENIR1.1	"1"	DDR23.2	"0"	PFR23.2	"1"	—	—
INT10 pin input	ENIR1.2	"1"	DDR23.4	"0"	PFR23.4	"1"	—	—
INT11 pin input	ENIR1.3	"1"	DDR23.6	"0"	PFR23.6	"1"	—	—
INT12 pin input	ENIR1.4	"1"	DDR22.0	"0"	PFR22.0	"1"	—	—
INT13 pin input	ENIR1.5	"1"	DDR22.2	"0"	PFR22.2	"1"	—	—
INT14 pin input	ENIR1.6	"1"	DDR22.4	"0"	PFR22.4	"1"	—	—
INT15 pin input	ENIR1.7	"1"	DDR22.6	"0"	PFR22.6	"1"	—	—
INT16 pin input	ENIR2.0	"1"	DDR27.0	"0"	PFR27.0, EPFR27.0	"1"	AD0ERH.16	"0"
INT17 pin input	ENIR2.1	"1"	DDR27.1	"0"	PFR27.1, EPFR27.1	"1"	AD0ERH.17	"0"
INT18 pin input	ENIR2.2	"1"	DDR27.2	"0"	PFR27.2, EPFR27.2	"1"	AD0ERH.18	"0"
INT19 pin input	ENIR2.3	"1"	DDR27.3	"0"	PFR27.3, EPFR27.3	"1"	AD0ERH.19	"0"
INT20 pin input	ENIR2.4	"1"	DDR27.4	"0"	PFR27.4, EPFR27.4	"1"	AD0ERH.20	"0"
INT21 pin input	ENIR2.5	"1"	DDR27.5	"0"	PFR27.5, EPFR27.5	"1"	AD0ERH.21	"0"
INT22 pin input	ENIR2.6	"1"	DDR27.6	"0"	PFR27.6, EPFR27.6	"1"	AD0ERH.22	"0"
INT23 pin input	ENIR2.7	"1"	DDR27.7	"0"	PFR27.7, EPFR27.7	"1"	AD0ERH.23	"0"
INT24 pin input	ENIR3.0	"1"	DDR26.0	"0"	PFR26.0, EPFR26.0	"1"	AD0ERH.24	"0"
INT25 pin input	ENIR3.1	"1"	DDR26.1	"0"	PFR26.1, EPFR26.1	"1"	AD0ERH.25	"0"
INT26 pin input	ENIR3.2	"1"	DDR26.2	"0"	PFR26.2, EPFR26.2	"1"	AD0ERH.26	"0"
INT27 pin input	ENIR3.3	"1"	DDR26.3	"0"	PFR26.3, EPFR26.3	"1"	AD0ERH.27	"0"
INT28 pin input	ENIR3.4	"1"	DDR26.4	"0"	PFR26.4, EPFR26.4	"1"	AD0ERH.28	"0"
INT29 pin input	ENIR3.5	"1"	DDR26.5	"0"	PFR26.5, EPFR26.5	"1"	AD0ERH.29	"0"
INT30 pin input	ENIR3.6	"1"	DDR26.6	"0"	PFR26.6, EPFR26.6	"1"	AD0ERH.30	"0"
INT31 pin input	ENIR3.7	"1"	DDR26.7	"0"	PFR26.7, EPFR26.7	"1"	AD0ERH.31	"0"

Note: Even though the external interrupt can be even used with setting DDR=0 and PFR=0 (general purpose port input mode), the input line will be disabled when setting STOP state with HIZ.

Note: To enable the external interrupt in STOP state, the port function registers (PFR, EPFR) must be set.

Note: For channels INT16 to INT31, the corresponding ADC channel enable bits (AD0ERH.ADEn, n=16 to 31) must be cleared additionally. Otherwise any digital input function is disabled.

25.7.3 What interrupt registers are used?

Setting of interrupt vectors of external interrupts, and interrupt levels

The relationship among external interrupt numbers, interrupt levels, and vectors is shown in the table below.

See “[Chapter 24 Interrupt Control \(Page No.429\)](#)” about the details of interrupt levels and interrupt vectors.

INT	Interrupt vectors (default)		Interrupt level setting bits (ICR[4:0])
	INT #	Address	
INT0	#16	0FFFBCh	Interrupt level register (ICR00) Address: 00440h
INT1	#17	0FFFB8h	
INT2	#18	0FFFB4h	Interrupt level register (ICR01) Address: 00441h
INT3	#19	0FFFB0h	
INT4	#20	0FFFACh	Interrupt level register (ICR02) Address: 00442h
INT5	#21	0FFFA8h	
INT6	#22	0FFFA4h	Interrupt level register (ICR03) Address: 00443h
INT7	#23	0FFFA0h	
INT8	#24	0FFF9Ch	Interrupt level register (ICR04) Address: 00444h
INT9	#25	0FFF98h	
INT10	#26	0FFF94h	Interrupt level register (ICR05) Address: 00445h
INT11	#27	0FFF90h	
INT12	#28	0FFF8Ch	Interrupt level register (ICR06) Address: 00446h
INT13	#29	0FFF88h	
INT14	#30	0FFF84h	Interrupt level register (ICR07) Address: 00447h
INT15	#31	0FFF80h	
INT16	#16	0FFFBCh	Interrupt level register (ICR00) Address: 00440h
INT17	#17	0FFFB8h	
INT18	#18	0FFFB4h	Interrupt level register (ICR01) Address: 00441h
INT19	#19	0FFFB0h	
INT20	#20	0FFFACh	Interrupt level register (ICR02) Address: 00442h
INT21	#21	0FFFA8h	
INT22	#22	0FFFA4h	Interrupt level register (ICR03) Address: 00443h
INT23	#23	0FFFA0h	
INT24	#24	0FFF9Ch	Interrupt level register (ICR04) Address: 00444h
INT25	#25	0FFF98h	
INT26	#26	0FFF94h	Interrupt level register (ICR05) Address: 00445h
INT27	#27	0FFF90h	
INT28	#28	0FFF8Ch	Interrupt level register (ICR06) Address: 00446h
INT29	#29	0FFF88h	
INT30	#30	0FFF84h	Interrupt level register (ICR07) Address: 00447h
INT31	#31	0FFF80h	

25.7.4 Interrupt types

Interrupt causes are limited to external interrupts. There is no bit for selection.

MB91460 Series**25.7.5 How to enable, disable, and clear interrupts**

Interrupt request enable bit, interrupt flag

Use interrupt request enable bits

- ENIR0.ENx. (x = 0-7)
- ENIR1.ENx. (x = 8-15)
- ENIR2.ENx. (x = 16-23)
- ENIR3.ENx. (x = 24-31)

to enable interrupts.

	Interrupt request enable bit (ENn [n = 0-15])
To disable interrupt requests	Set to "0"
To enable interrupt requests	Set to "1"

Use interrupt flags

- EIRR0.ERx. (x = 0-7)
- EIRR1.ERx. (x = 8-15)
- EIRR2.ERx. (x = 16-23)
- EIRR3.ERx. (x = 24-31)

to clear interrupt requests.

	Interrupt flag (ERn [n = 0-31])
To clear interrupts	Write "0"

25.8. Caution

- When the External interrupt request level is set to “H” level or “L” level (LAn, LBn = “00” or “01”) and when the INT pin input becomes activated, the corresponding flag (ERn) will be re-set to “1” even if the external interrupt flag (ERn) is set to “0” by software simultaneously.

Note: n = 0 to 31

- Before enabling the external interrupt request with ENn = “1”, it is recommended to clear the external interrupt flag (set ERn to “0”) to avoid interrupts caused by previous matches of the input trigger (the ERn flag is set independently of the setting of ENn).

Note: n = 0 to 31

- Before going into standby (STOP state), make sure to disable unused external interrupts (ENn = “0”).

Note: n = 0 - 31

- Minimum 2 x CLKP (peripheral clock) is required for the pulse width to detect a valid interrupt level or edge.
- When waking up from STOP state a minimum pulse width (> 50ns) of the INT signal trigger must be fulfilled.
- The interrupt request to the interrupt controller remains active, even if an external interrupt is input from the external interrupt pin INTn and canceled afterward, since the interrupt flag (ERn) is present. To cancel the interrupt request to the interrupt controller, the interrupt flag must be cleared (ERn = “0”) by software. (See the diagram in “[25.5. Operation \(Page No.457\)](#)”)

Note: n = 0 to 31

Chapter 26 DMA Controller

26.1. Overview of the DMA Controller (DMAC)

The DMA controller (DMAC) is a module that implements DMA (Direct Memory Access) transfer on FR family devices. When this module is used to control DMA transfer, various kinds of data can be transferred at high speed by bypassing the CPU, enhancing system performance.

26.1.1 Hardware Configuration

The DMA controller (DMAC) consists mainly of the following blocks:

Five independent DMA channels

- 5-channel independent access control circuit
- 32-bit address registers (reload specifiable, two registers for each channel)
- 16-bit transfer count register (reload specifiable, one register for each channel)
- 4-bit block count register (one for each channel)
- Up to 128 internal transfer request sources
- External transfer request input pins: DREQ0, DREQ1, DREQ2, DREQ3 (for ch0-3 only)
- External transfer request acknowledgement output pins: DACKX0, DACKX1, DACKX2, DACKX3 (for ch0-3 only)
- DMA end output pins: DEOP0, DEOP1, DEOP2, DEOP3 (for ch0-3 only)
- External transfer stop: DEOTX0, DEOTX1, DEOTX2, DEOTX3 (for ch0-3 only)
- Fly-by transfer (external memory to external I/O and external I/O to external memory) (for ch0-3 only)
- 2-cycle transfer

26.1.2 Main Functions

The following are the main functions related to data transfer by the DMA controller (DMAC):

Data can be transferred independently over multiple channels (5 channels)

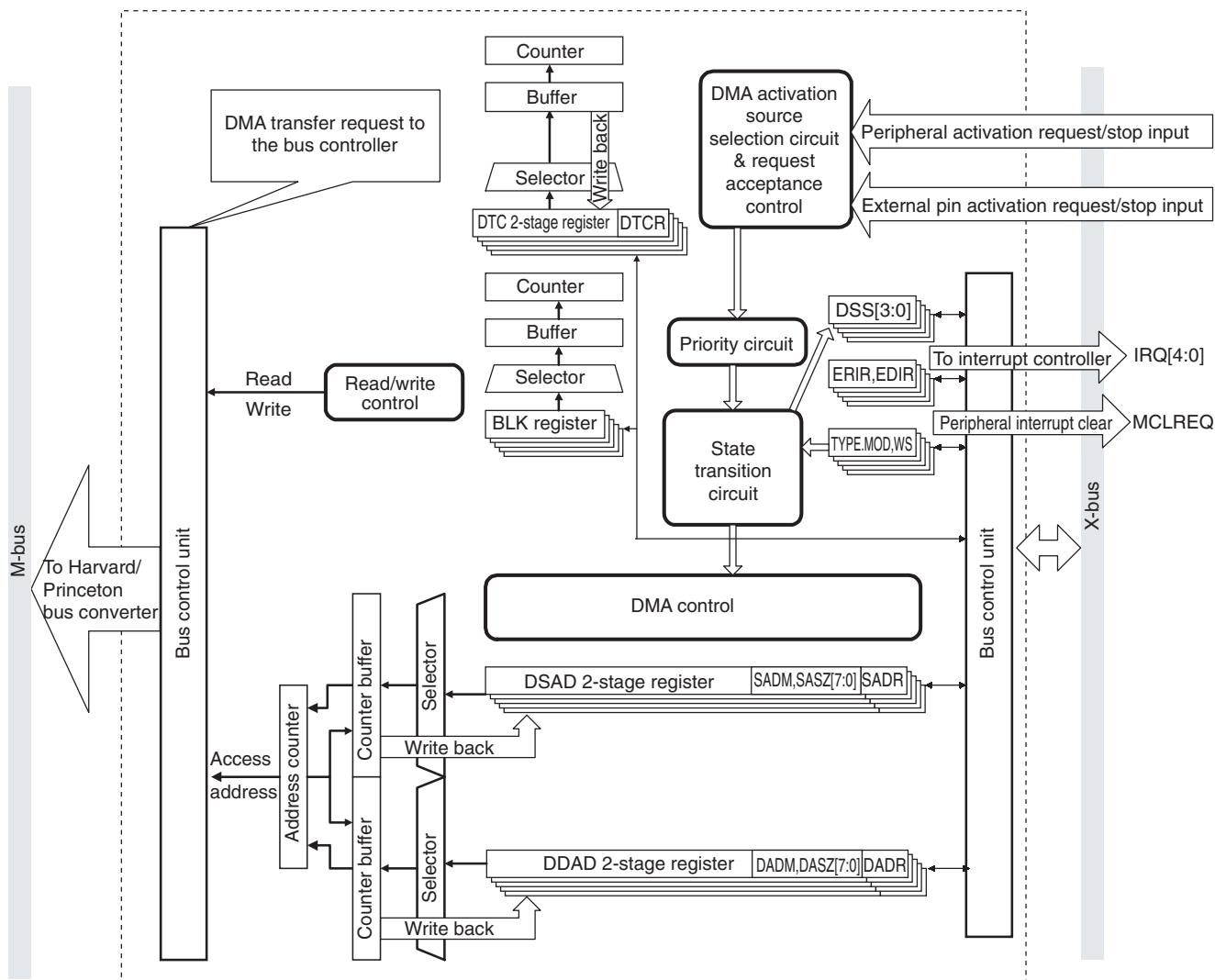
- Priority (ch.0>ch.1>ch.2>ch.3>ch.4)
- The priority order can be rotated between ch.0 and ch.1.
- Transfer request sources
 - External dedicated pin input (edge detection/level detection for ch0-3 only)
 - Built-in peripheral requests (shared interrupt requests, including external interrupts)
 - Software request (register write)
- Transfer mode
 - Demand transfer, burst transfer, step transfer, and block transfer
 - Addressing mode: 32-bit full addressing (increment/decrement/fixed)
The address increment/decrement range is from -255 to +255.
 - Data types: Byte, halfword, and word length

- Single shot/reload selectable

26.1.3 Block Diagram

Figure 26.1-1 "Block Diagram of the DMA Controller (DMAC)" is a block diagram of the DMA controller (DMAC).

Figure 26.1-1 Block Diagram of the DMA Controller (DMAC)



26.2. DMA Controller (DMAC) Registers

This section describes the configuration and functions of the registers used by the DMA controller (DMAC).

26.2.1 DMA Controller (DMAC) registers

Figure 26.2-1 "DMA Controller (DMAC) Registers" shows the registers of the DMA controller (DMAC).

Figure 26.2-1 DMA Controller (DMAC) Registers

(bit)	31	24	23	16	15	08	07	00		
ch.0									Control/status register A	(DMACA0)
ch.0									Control/status register B	(DMACB0)
ch.1									Control/status register A	(DMACA1)
ch.1									Control/status register B	(DMACB1)
ch.2									Control/status register A	(DMACA2)
ch.2									Control/status register B	(DMACB2)
ch.3									Control/status register A	(DMACA3)
ch.3									Control/status register B	(DMACB3)
ch.4									Control/status register A	(DMACA4)
ch.4									Control/status register B	(DMACB4)
									All-channel control register	(DMACR)
ch.0									Transfer source address register	(DMASA0)
ch.0									Transfer destination address register	(DMADA0)
ch.1									Transfer source address register	(DMASA1)
ch.1									Transfer destination address register	(DMADA1)
ch.2									Transfer source address register	(DMASA2)
ch.2									Transfer destination address register	(DMADA2)
ch.3									Transfer source address register	(DMASA3)
ch.3									Transfer destination address register	(DMADA3)
ch.4									Transfer source address register	(DMASA4)
ch.4									Transfer destination address register	(DMADA4)

26.2.2 Notes on Setting Registers

When the DMA controller (DMAC) is set, some bits need to be set while DMA is stopped. If they are set while DMA is in progress (during transfer), correct operation cannot be guaranteed.

An asterisk (*) following a bit when its function is described later indicates that the operation of the bit is affected if it is set during DMAC transfer. Rewrite this bit while DMAC transfer is stopped (start is disabled or temporarily stopped).

If such a bit is set while DMA transfer start is disabled (when DMAE of DMACR=0, or DENB of DMACA=0), the setting takes effect when start is enabled.

If such a bit is set while DMA transfer is temporarily stopped (DMAH[3:0] of DMACR not equal to 0000_B or PAUS of DMACA=1), the setting takes effect when temporary stopping is canceled.

26.2.3 Control/Status Registers A (DMACA0 to 4)

Control/status registers A (DMACA0 to 4) control the operation of the DMAC channels. There is a separate register for each channel.

This section describes the configuration and functions of control/status registers A (DMACA0 to 4).

■ Bit Configuration of Control/Status Registers A (DMACA0 to 4)

Figure 26.2-2 "Bit Configuration of Control/Status Registers A (DMACA0 to 4)" shows the bit configuration of control/status registers A (DMACA0 to 4).

Figure 26.2-2 Bit Configuration of Control/Status Registers A (DMACA0 to 4)

Address	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Initial value
000200 _H (ch0)		DENB	PAUS	STRG														000000000000XXXX
000208 _H (ch1)																		
000210 _H (ch2)	bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000218 _H (ch3)																		XXXXXXXXXXXXXXXX _B
000220 _H (ch4)																		

■ Detailed Bit of Control/Status Registers A (DMACA0 to 4)

The following describes the functions of the bits of control/status registers A (DMACA0 to 4).

[Bit 31] DENB (Dma ENaBle): DMA operation enable bit

This bit, which corresponds to a transfer channel, is used to enable and disable DMA transfer.

The activated channel starts DMA transfer when a transfer request is generated and accepted.

All transfer requests that are generated for a deactivated channel are disabled.

When the transfer on an activated channel reaches the specified count, this bit is set to 0 and transfer stops.

The transfer can be forced to stop by writing 0 to this bit. Be sure to stop a transfer forcibly (0 write) only after temporarily stopping DMA using the PAUS bit (Bit30 of DMACA). If the transfer is forced to stop without first temporarily stopping DMA, DMA stops but the transferred data cannot be guaranteed. Check whether DMA is stopped using the DSS[2:0] bits [Bit18-16 of DMACB].

DENB	Function
0	Disables operation of DMA on the corresponding channel (initial value).
1	Enables operation of DMA on the corresponding channel.

- If a stop request is accepted during reset: Initialized to 0.
- This bit is readable and writable.
- If the operation of all channels is disabled by Bit15 (DMAE bit) of the DMAC all-channel control register (DMACR), writing 1 to this bit is disabled and the stopped state is maintained. If the operation is disabled by the above bit while it is enabled by this bit, 0 is written to this bit and the transfer is stopped (forced stop).

[Bit 30] PAUS (PAUSE)*: Temporary stop instruction

This bit temporarily stops DMA transfer on the corresponding channel. If this bit is set, DMA transfer is not performed before this bit is cleared (While DMA is stopped, the DSS bits are 1xx_B).

If this bit is set before starting, DMA transfer continues to be temporarily stopped.

New transfer requests that occur while this bit is set are accepted, but no transfer starts before this bit is cleared (See 26.3.9"Operation from Starting to End/Stopping").

PAUS	Function
0	Enables operation of the corresponding channel DMA (initial value)
1	Temporarily stops DMA on the corresponding channel.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 29] STRG (Software TRiGger): Transfer request

This bit generates a DMA transfer request for the corresponding channel. If 1 is written to this bit, a transfer request is generated when write operation to the register is completed and transfer on the corresponding channel is started.

However, if the corresponding channel is not activated, operations on this bit are disabled.

If starting by a write operation to the DMAE bit and a transfer request occurring due to this bit are simultaneous, the transfer request is enabled and transfer is started. If writing of 1 to the PAUS bit and a transfer request occurring due to this bit are simultaneous, the transfer request is enabled, but DMA transfer is not started before 0 is written to the PAUS bit.

STRG	Function
0	Disabled
1	DMA starting request

- When reset: Initialized to 0.
- The read value is always 0.
- Only a write value of 1 is valid. If 0 is write, operation is not affected.

[Bits 28 to 24] IS4 to 0 (Input Select)*: Transfer source selection

These bits select the source of a transfer request. Note that the software transfer request by the STRG bit function is always valid regardless of the setting of these bits. As listed in [Table 26.2-1](#) "Settings for Transfer Request Sources".

Table 26.2-1 Settings for Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request
00000	-	-	Activation by hardware prohibited	not available
00001	-	-	Setting prohibited	
-			-	
01101			Setting prohibited	
01110	-	-	External DMA-pin high level or rising edge	
01111	-	-	External DMA-pin low level or falling edge	
10000	0000	0	External Interrupt 0	-
10001	0000	1	External Interrupt 1	-
10010	0000	2	External Interrupt 2	-
10011	0000	3	External Interrupt 3	-
10100	0000	4	Reload Timer 0	-
10101	0000	5	Reload Timer 1	-
10110	0000	6	USART (LIN) 0 RX	available
10111	0000	7	USART (LIN) 0 TX	-
11000	0000	8	USART (LIN) 1 RX	available
11001	0000	9	USART (LIN) 1 TX	-
11010	0000	10	USART (LIN, FIFO) 4 RX	available
11011	0000	11	USART (LIN, FIFO) 4 TX	-
11100	0000	12	USART (LIN, FIFO) 5 RX	available
11101	0000	13	USART (LIN, FIFO) 5 TX	-
11110	0000	14	A/D Converter	-
11111	0000	15	Programmable Pulse Generator (PPG) 0	-

- When reset: IS4-0 is initialized to 00000_B.
- When reset: EIS3-0 is initialized to 0000_B.
- These bits are readable and writable.

MB91460 Series**Notes:**

- If DMA start resulting from an interrupt from a peripheral function is set (IS=1xxxx_B), disable interrupts from the selected peripheral function with the ICR register.
- If demand transfer mode is selected, only IS[4:0]=01110_B, 01111_B can be set. Starting by other sources is disabled.
- External request input is valid only for CH0, 1, and 2. External request input cannot be selected for CH2, CH3 and 4. Whether level detection or edge detection is used is determined by the mode setting. Level detection is selected for demand transfer. For all other cases, edge detection is selected.

[Bits 23 to 20] EIS3 to 0 (Extended Input Select)*: Extended Transfer Source Selection

These bits select the source of a transfer request note that the software transfer request by the STRG bit function is always valid regardless of the setting of these bits. As listed in [Table 26.2-2](#) "Settings for Extended Transfer Request Sources".

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10000	0001	16	External Interrupt 0	-	
10001	0001	17	External Interrupt 1	-	
10010	0001	18	External Interrupt 2	-	
10011	0001	19	External Interrupt 3	-	
10100	0001	20	External Interrupt 4	-	
10101	0001	21	External Interrupt 5	-	
10110	0001	22	External Interrupt 6	-	
10111	0001	23	External Interrupt 7	-	
11000	0001	24	Up/Down Counter 0	-	not available on MB91V460A
11001	0001	25	Up/Down Counter 1	-	not available on MB91V460A
11010	0001	26	Up/Down Counter 2	-	not available on MB91V460A
11011	0001	27	Up/Down Counter 3	-	not available on MB91V460A
11100	0001	28	I ² C 0	available	not available on MB91V460A
11101	0001	29	I ² C 1	-	not available on MB91V460A
11110	0001	30	I ² C 2	-	not available on MB91V460A
11111	0001	31	I ² C 3	-	not available on MB91V460A
10000	0010	32	Reload Timer 0	-	
10001	0010	33	Reload Timer 1	-	
10010	0010	34	Reload Timer 2	-	
10011	0010	35	Reload Timer 3	-	

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10100	0010	36	Reload Timer 4	-	
10101	0010	37	Reload Timer 5	-	
10110	0010	38	Reload Timer 6	-	
10111	0010	39	Reload Timer 7	-	
11000	0010	40	Free-Run Timer 0	-	
11001	0010	41	Free-Run Timer 1	-	
11010	0010	42	Free-Run Timer 2	-	
11011	0010	43	Free-Run Timer 3	-	
11100	0010	44	Free-Run Timer 4	-	
11101	0010	45	Free-Run Timer 5	-	
11110	0010	46	Free-Run Timer 6	-	
11111	0010	47	Free-Run Timer 7	-	
10000	0011	48	USART (LIN) 0 RX	available	
10001	0011	49	USART (LIN) 0 TX	-	
10010	0011	50	USART (LIN) 1 RX	available	
10011	0011	51	USART (LIN) 1 TX	-	
10100	0011	52	USART (LIN) 2 RX	available	
10101	0011	53	USART (LIN) 2 TX	-	
10110	0011	54	USART (LIN) 3 RX	available	
10111	0011	55	USART (LIN) 3 TX	-	
11000	0011	56	USART (LIN, FIFO) 4 RX	available	
11001	0011	57	USART (LIN, FIFO) 4 TX	-	
11010	0011	58	USART (LIN, FIFO) 5 RX	available	
11011	0011	59	USART (LIN, FIFO) 5 TX	-	
11100	0011	60	USART (LIN, FIFO) 6 RX	available	
11101	0011	61	USART (LIN, FIFO) 6 TX	-	
11110	0011	62	USART (LIN, FIFO) 7 RX	available	
11111	0011	63	USART (LIN, FIFO) 7 TX	-	
10000	0100	64	USART (LIN) 8 RX	available	
10001	0100	65	USART (LIN) 8 TX	-	
10010	0100	66	USART (LIN) 9 RX	available	

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10011	0100	67	USART (LIN) 9 TX	-	
10100	0100	68	USART (LIN) 10 RX	available	
10101	0100	69	USART (LIN) 10 TX	-	
10110	0100	70	USART (LIN) 11 RX	available	
10111	0100	71	USART (LIN) 11 TX	-	
11000	0100	72	USART (LIN) 12 RX	available	
11001	0100	73	USART (LIN) 12 TX	-	
11010	0100	74	USART (LIN) 13 RX	available	
11011	0100	75	USART (LIN) 13 TX	-	
11100	0100	76	USART (LIN) 14 RX	available	
11101	0100	77	USART (LIN) 14 TX	-	
11110	0100	78	USART (LIN) 15 RX	available	
11111	0100	79	USART (LIN) 15 TX	-	
10000	0101	80	Input Capture 0	-	
10001	0101	81	Input Capture 1	-	
10010	0101	82	Input Capture 2	-	
10011	0101	83	Input Capture 3	-	
10100	0101	84	Input Capture 4	-	
10101	0101	85	Input Capture 5	-	
10110	0101	86	Input Capture 6	-	
10111	0101	87	Input Capture 7	-	
11000	0101	88	Output Compare 0	-	
11001	0101	89	Output Compare 1	-	
11010	0101	90	Output Compare 2	-	
11011	0101	91	Output Compare 3	-	
11100	0101	92	Output Compare 4	-	
11101	0101	93	Output Compare 5	-	
11110	0101	94	Output Compare 6	-	
11111	0101	95	Output Compare 7	-	
10000	0110	96	Programmable Pulse Generator 0	-	
10001	0110	97	Programmable Pulse Generator 1	-	

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10010	0110	98	Programmable Pulse Generator 2	-	
10011	0110	99	Programmable Pulse Generator 3	-	
10100	0110	100	Programmable Pulse Generator 4	-	
10101	0110	101	Programmable Pulse Generator 5	-	
10110	0110	102	Programmable Pulse Generator 6	-	
10111	0110	103	Programmable Pulse Generator 7	-	
11000	0110	104	Programmable Pulse Generator 8	-	
11001	0110	105	Programmable Pulse Generator 9	-	
11010	0110	106	Programmable Pulse Generator 10	-	
11011	0110	107	Programmable Pulse Generator 11	-	
11100	0110	108	Programmable Pulse Generator 12	-	
11101	0110	109	Programmable Pulse Generator 13	-	
11110	0110	110	Programmable Pulse Generator 14	-	
11111	0110	111	Programmable Pulse Generator 15	-	
10000	0111	112	ADC 0	-	
10001	0111	113	ADC 1	-	not available on MB91V460A
10010	0111	114	Phase Frequency Modulator	-	not available on MB91V460A
10011	0111	115	reserved		
10100	0111	116	FlexRay Channel 0 IBF	-	not available on MB91V460A
10101	0111	117	FlexRay Channel 0 OBF	-	not available on MB91V460A
10110	0111	118	FlexRay Channel 1 IBF	-	not available on MB91V460A
10111	0111	119	FlexRay Channel 1 OBF	-	not available on MB91V460A
11000	0111	120	ADC 0 End of Scan		not available on MB91V460A
11001	0111	121	ADC 1 End of Scan		not available on MB91V460A
11010	0111	122	reserved		
11011	0111	123	reserved		
11100	0111	124	reserved		
11101	0111	125	I ² S Even Channels Tx/Rx	-	not available on MB91V460A
11110	0111	126	I ² S Odd Channels Tx/Rx	-	not available on MB91V460A
11111	0111	127	reserved		
10000	1000	128	Reload Timer 8	-	not available on MB91V460A

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10001	1000	129	Reload Timer 9	-	not available on MB91V460A
10010	1000	130	Reload Timer 10	-	not available on MB91V460A
10011	1000	131	Reload Timer 11	-	not available on MB91V460A
10100	1000	132	Reload Timer 11	-	not available on MB91V460A
10101	1000	133	Reload Timer 13	-	not available on MB91V460A
10110	1000	134	Reload Timer 14	-	not available on MB91V460A
10111	1000	135	Reload Timer 15	-	not available on MB91V460A
11000	1000	136	External Interrupt 16	-	not available on MB91V460A
11001	1000	137	External Interrupt 17	-	not available on MB91V460A
11010	1000	138	External Interrupt 18	-	not available on MB91V460A
11011	1000	139	External Interrupt 19	-	not available on MB91V460A
11100	1000	140	External Interrupt 20	-	not available on MB91V460A
11101	1000	141	External Interrupt 21	-	not available on MB91V460A
11110	1000	142	External Interrupt 22	-	not available on MB91V460A
11111	1000	143	External Interrupt 23	-	not available on MB91V460A
10000	1001	144	Programmable Pulse Generator 16	-	not available on MB91V460A
10001	1001	145	Programmable Pulse Generator 17	-	not available on MB91V460A
10010	1001	146	Programmable Pulse Generator 18	-	not available on MB91V460A
10011	1001	147	Programmable Pulse Generator 19	-	not available on MB91V460A
10100	1001	148	Programmable Pulse Generator 20	-	not available on MB91V460A
10101	1001	149	Programmable Pulse Generator 21	-	not available on MB91V460A
10110	1001	150	Programmable Pulse Generator 22	-	not available on MB91V460A
10111	1001	151	Programmable Pulse Generator 23	-	not available on MB91V460A
11000	1001	152	Programmable Pulse Generator 24	-	not available on MB91V460A
11001	1001	153	Programmable Pulse Generator 25	-	not available on MB91V460A
11010	1001	154	Programmable Pulse Generator 26	-	not available on MB91V460A
11011	1001	155	Programmable Pulse Generator 27	-	not available on MB91V460A
11100	1001	156	Programmable Pulse Generator 28	-	not available on MB91V460A
11101	1001	157	Programmable Pulse Generator 29	-	not available on MB91V460A
11110	1001	158	Programmable Pulse Generator 30	-	not available on MB91V460A
11111	1001	159	Programmable Pulse Generator 31	-	not available on MB91V460A

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
10000	1010	160	APIX [®] Transaction Buffer 00	available	not available on MB91V460A
10001	1010	161	APIX [®] Transaction Buffer 01	available	not available on MB91V460A
10010	1010	162	APIX [®] Transaction Buffer 02	available	not available on MB91V460A
10011	1010	163	APIX [®] Transaction Buffer 03	available	not available on MB91V460A
10100	1010	164	APIX [®] Transaction Buffer 04	available	not available on MB91V460A
10101	1010	165	APIX [®] Transaction Buffer 05	available	not available on MB91V460A
10110	1010	166	APIX [®] Transaction Buffer 06	available	not available on MB91V460A
10111	1010	167	APIX [®] Transaction Buffer 07	available	not available on MB91V460A
11000	1010	168	APIX [®] Transaction Buffer 08	available	not available on MB91V460A
11001	1010	169	APIX [®] Transaction Buffer 09	available	not available on MB91V460A
11010	1010	170	APIX [®] Transaction Buffer 10	available	not available on MB91V460A
11011	1010	171	APIX [®] Transaction Buffer 11	available	not available on MB91V460A
11100	1010	172	APIX [®] Transaction Buffer 12	available	not available on MB91V460A
11101	1010	173	APIX [®] Transaction Buffer 13	available	not available on MB91V460A
11110	1010	174	APIX [®] Transaction Buffer 14	available	not available on MB91V460A
11111	1010	175	APIX [®] Transaction Buffer 15	available	not available on MB91V460A
10000	1011	176	Free-Run Timer 8	-	not available on MB91V460A
10001	1011	177	Free-Run Timer 9	-	not available on MB91V460A
10010	1011	178	Free-Run Timer 10	-	not available on MB91V460A
10011	1011	179	Free-Run Timer 11	-	not available on MB91V460A
10100	1011	180	Input Capture 8	-	not available on MB91V460A
10101	1011	181	Input Capture 9	-	not available on MB91V460A
10110	1011	182	reserved		
10111	1011	183	reserved		
11000	1011	184	reserved		
11001	1011	185	reserved		
11010	1011	186	reserved		
11011	1011	187	reserved		

Table 26.2-2 Settings for Extended Transfer Request Sources

IS	EIS	RN	Function	Transfer stop request	Comments
11100	1011	188	I ² C 4	available	not available on MB91V460A
11101	1011	189	I ² C 5	available	not available on MB91V460A
11110	1011	190	I ² C 6	available	not available on MB91V460A
11111	1011	191	I ² C 7	available	not available on MB91V460A
10000	1100	192	reserved	-	
10001	1100	193	reserved		
10010	1100	194	reserved		
10011	1100	195	Data Flash Write Sequencer	-	not available on MB91V460A
10100	1100	196	reserved		
10101	1100	197	reserved		
10110	1100	198	reserved		
10111	1100	199	Programmable CRC Module	-	not available on MB91V460A
11000	1100	200	USB Endpoint 0 IN (IRQ0)	-	not available on MB91V460A
11001	1100	201	USB Endpoint 0 OUT (IRQ1)	-	not available on MB91V460A
11010	1100	202	USB Endpoint 1 (IRQ2)	-	not available on MB91V460A
11011	1100	203	USB Endpoint 2 (IRQ3)	-	not available on MB91V460A
11100	1100	204	USB Endpoint 3 (IRQ4)	-	not available on MB91V460A
11101	1100	205	USB Endpoint 4 (IRQ5)	-	not available on MB91V460A
11110	1100	206	USB Endpoint 5 (IRQ6)	-	not available on MB91V460A
11111	1100	207	reserved		

- When reset: IS4-0 is initialized to 00000_B.
- When reset: EIS3-0 is initialized to 0000_B.
- These bits are readable and writable.

[Bits 19 to 16] BLK3 to 0 (BLoK size): Block size specification

These bits specify the block size for block transfer on the corresponding channel. The value specified by these bits becomes the number of words in one transfer unit (more exactly, the repetition count of the data width setting). If block transfer will not be performed, set 01_H (size 1). This register value is ignored during demand transfer. The size becomes 1.

BLK	Function
XXXX _B	Block size of the corresponding channel

- When reset: Not initialized.
- These bits are readable and writable.
- If 0 is specified for all bits, the block size becomes 16 words.
- During reading, the block size is always read (reload value).

[Bits 15 to 00] DTC (Dma Terminal Count register)*: Transfer count register

The DTC register stores the transfer count. Each register has 16-bit length.

All registers have a dedicated reload register. When the register is used for a channel that is enabled to reload the transfer count register, the initial value is automatically written back to the register when the transfer is completed.

DTC	Function
XXXX _B	Transfer count for the corresponding channel

When DMA transfer is started, the data from this register is stored in the counter buffer of the DMA-dedicated transfer counter and is decremented by 1 (subtraction) after each transfer unit. When DMA transfer is completed, the contents of the counter buffer are written back to this register and then DMA ends. Thus, the transfer count value during DMA operation cannot be read.

- When reset: Not initialized.
- These bits are readable and writable. Always access DTC using halfword length or word length.
- During reading, the count value is read. The reload value cannot be read.

26.2.4 Control/Status Registers B (DMACB0 to 4)

Control/status registers B (DMACB0 to 4) control the operation of each DMAC channel and exist separately for each channel.

This section describes the configuration of control/status registers B (DMACB0 to 4) and their functions.

■ Bit Configuration of Control/Status Register B (DMACB0 to 4)

Figure 26.2-3 "Bit Configuration of Control/Status Registers B (DMACB0 to 4)" shows the bit configuration of control/status registers B (DMACB0 to 4).

Figure 26.2-3 Bit Configuration of Control/Status Registers B (DMACB0 to 4)

Address	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Initial value	
000204 _H (ch0)																			
00020C _H (ch1)		TYPE[1:0]		MOD[1:0]		WS[1:0]		SADM	DADM	DTCR	SADR	DADR	ERIE	EDIE	DSS[2:0]			0000000000000000	
000214 _H (ch2)	bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	XXXXXXXXXXXXXXXX _B	
00021C _H (ch3)		SASZ[7:0]								DASZ[7:0]									
000224 _H (ch4)		SASZ[7:0]								DASZ[7:0]									

■ Detailed Bit of Control/Status Register B (DMACB0 to 4)

The following describes the functions of the bits of control status register B (DMACB0 to 4).

[Bits 31 to 30] TYPE (TYPE)*: Transfer type setting

These bits are the transfer type setting bits and set the type of operation for the corresponding channel.

- 2-cycle transfer type: In this type, the transfer source address (DMASA) and transfer destination address

MB91460 Series

(DMADA) are set and transfer is performed by repeating the read operation and write operation for the number of times specified by the transfer counter buffer. All areas can be specified as a transfer source or transfer destination (32-bit address).

- Fly-by transfer type: In this type, external <--> external transfer is performed in one cycle by setting a external memory address as the transfer destination address (DMADA). Be sure to specify an external area for the memory address.

Table 26.2-3 Settings for the Transfer Types

TYPE	Function
00 _B	2-cycle transfer (initial value)
01 _B	Fly-by: External Memory --> external I/O transfer
10 _B	Fly-by: external I/O --> external memory transfer
11 _B	Setting disabled

- When reset: Initialized to 00_B.
- These bits are readable and writable.

[Bits 29, 28] MOD (MODE)*: Transfer mode setting

These bits are the transfer mode setting bits and set the operating mode of the corresponding channel.

Table 26.2-4 Settings for Transfer Modes

MOD	Function
00 _B	Block/step transfer mode (initial value)
01 _B	Burst transfer mode
10 _B	Demand transfer mode
11 _B	Setting disabled

- When reset: Initialized to 00_B.
- These bits are readable and writable.

[Bits 27 to 26] WS (Word Size)*: Transfer data width selection

These bits are the transfer data width selection bits and are used to select the transfer data width of the corresponding channel. Transfer operations are repeated in units of the data width specified in this register for as many times as the specified count.

Table 26.2-5 Selection of the Transfer Data Width

WS	Function
00 _B	Byte-width transfer (initial value)
01 _B	Halfword-width transfer
10 _B	Word-width transfer
11 _B	Setting disabled

- When reset: Initialized to 00_B.
- These bits are readable and writable.

[Bit 25] SADM (Source-ADdr. Count-Mode select)*: Transfer source address count mode specification

This bit specifies the address processing of the transfer source address of the corresponding channel in each transfer operation.

An address increment is added or an address decrement is subtracted after each transfer operation according to the specified transfer source address count width (SASZ). When the transfer is completed, the next access address is written to the corresponding address register (DMASA).

As a result, the transfer source address register is not updated until DMA transfer is completed.

To make the address always the same, specify 0 or 1 for this register and make the address count width (SASZ and DASZ) equal to 0.

SADM	Function
0	Increments transfer source address. (initial value)
1	Decrements the transfer source address.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 24] DADM (Destination-ADdr. Count-Mode select)*: Transfer destination address count mode specification

This bit specifies the address processing for the transfer destination address of the corresponding channel in each transfer operation.

An address increment is added or an address decrement is subtracted after each transfer operation according to the specified transfer destination address count width (DASZ). When the transfer is completed, the next access address is written to the corresponding address register (DMADA).

As a result, the transfer destination address register is not updated until the DMA transfer is completed.

To make the address always the same, specify 0 or 1 for this register and make the address count width (SASZ, DASZ) equal to 0.

DADM	Function
0	Increments the transfer source address. (initial value)
1	Decrements the transfer source address.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 23] DTCR (DTC-reg. Reload)*: Transfer count register reload specification

This bit controls reloading of the transfer count register for the corresponding channel.

If reload operation is enabled by this bit, the count register value is restored to its initial value after the transfer is completed then DMAC stops and then waiting starts for new transfer requests (an activation request by STRG or IS setting). If this bit is 1, the DENB bit is not cleared.

DENB=0 or DMAE=0 must be set to stop the transfer. In either case, the transfer is forcibly stopped.

If reloading of the counter is disabled, a single shot operation occurs. In single shot operation, operation stops after the transfer is completed even if reload is specified in the address register. The DENB bit is also cleared in this case.

DTCR	Function
0	Disables transfer count register reloading (initial value)
1	Enables transfer count register reloading.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 22] SADR (Source-ADdr.-reg. Reload)*: Transfer source address register reload specification

This bit controls reloading of the transfer source address register for the corresponding channel.

If this bit enables the reload operation, the transfer source address register value is restored to its initial value after the transfer is completed.

If reloading of the counter is disabled, a single shot operation occurs. In single shot operation, operation stops after the transfer is completed even if reload is specified in the address register. The address register value also stops in this case while the initial value is being reloaded.

If this bit disables the reload operation, the address register value when the transfer is completed is the address to be accessed next to the final address. When address increment is specified, the next address is an incremented address.

SADR	Function
0	Disables transfer source address register reloading. (initial value)
1	Enables transfer source address register reloading.

- When reset: Initialized to 0.
- This bit is readable and writable.

MB91460 Series**[Bit 21] DADR (Dest.-ADdr.-reg. Reload)*: Transfer destination address register reload specification**

This bit controls reloading of the transfer destination address register for the corresponding channel.

If this bit enables reloading, the transfer destination address register value is restored to its initial value after the transfer is completed.

The details of other functions are the same as those described for Bit22 (SADR).

DADR	Function
0	Disables transfer destination address register reloading. (initial value)
1	Enables transfer destination address register reloading.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 20] ERIE (ERror Interrupt Request Enable)*: Error interrupt Request output enable

This bit controls the occurrence of an interrupt for termination after an error occurs. The nature of the error that occurred is indicated by DSS2 to 0. Note that an interrupt occurs only for specific termination causes and not for all termination causes (Refer to bits DSS2 to 0, which are Bits 18 to 16).

ERIE	Function
0	Disables error interrupt request output. (initial value)
1	Enables error interrupt request output.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 19] EDIE (EnD Interrupt Request Enable)*: End interrupt request output enable

This bit controls the occurrence of an interrupt for normal termination.

EDIE	Function
0	Disables end interrupt request output. (initial value)
1	Enables end interrupt request output.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bits 18 to 16] DSS2 to 0 (DMA Stop Status)*: Transfer stop source indication

These bits indicate code (end code) of 3 bits that indicates the source of stopping or termination of DMA transfer on the corresponding channel. For a list of end codes, see [Table 26.2-6](#) and [Table 26.2-7](#) "End Codes".

Table 26.2-6 End Codes-1

DSS[2]	Function	Interrupt
0 _B	Initial value	None
1 _B	DMA stopped temporarily (for example, due to DMAH, PAUS bit, and an interrupt)	None

Table 26.2-7 End Codes-2

DSS[1:0]	Function	Interrupt
00 _B	Initial value	None
01 _B	Address error (underflow/overflow)	Error interrupt (ERIE)
10 _B	Transfer stop request	Error interrupt (ERIE)
11 _B	Normal end	End interrupt (EDIE)

A transfer stop request is set only when it is requested by a peripheral device or the external pin DSTP function is used.

The Interrupt column indicates the type of interrupts that can occur.

- When reset: Initialized to 000_B.
- These bits can be cleared by writing 000_B to them.
- These bits are readable and writable. Note that the only valid written value is 000.

[Bits 15 to 8] SASZ (Source Addr count SiZe)*: Transfer source address count size specification

These bits specify the increment or decrement width for the transfer source address (DMASA) of the corresponding channel in each transfer operation. The value set by these bits becomes the address increment/decrement for each transfer unit. The address increment/decrement conforms to the instruction in the transfer source address count mode (SADM).

SASZ	Function
XX _H	Specify the increment/decrement width of the transfer source address. 0 to 255

- When reset: Not initialized
- These bits are readable and writable.

MB91460 Series**[Bits 7 to 0] DASZ (Des Addr count SiZe)*: Transfer destination address count size specification**

These bits specify the increment or decrement width for the transfer destination address (DMADA) of the corresponding channel in each transfer operation. The value set by these bits becomes the address increment/decrement for each transfer unit. The address increment/decrement conforms to the instruction in the transfer destination address count mode (DADM).

DASZ	Function
XX _H	Specify the increment/decrement width of the transfer destination address. 0 to 255

- When reset: Not initialized
- These bits are readable and writable.

26.2.5 Transfer Source/Transfer Destination Address Setting Registers (DMASA0 to 4/DMADA0 to 4)

The transfer source/transfer destination address setting registers (DMASA0 to 4/DMADA0 to 4) control the operation of the DMAC channels. These registers are separately existent for each channel.

This section describes the configuration and functions of the transfer source/transfer destination address setting registers (DMASA0 to 4/DMADA0 to 4).

■ Bit Configuration of Transfer Source/Transfer Destination Address Setting Registers (DMASA0 to 4/DMADA0 to 4)

The transfer source/transfer destination address setting registers (DMASA0 to 4/DMADA0 to 4) are a group of registers that store the transfer source/transfer destination addresses. Each register is 32 bits length.

Figure 26.2-4 "Bit Configuration of the Transfer Source/Transfer Destination Address Setting Registers" shows the bit configuration of the transfer source/transfer destination address setting registers (DMASA0 to 4/DMADA0 to 4).

Figure 26.2-4 Bit Configuration of the Transfer Source/Transfer Destination Address Setting Registers

Address	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Initial value
001000 _H (ch0)	bit	DMASA[31:16]																XXXXXXXXXXXXXXXX
001008 _H (ch1)																		
001010 _H (ch2)		DMASA[15:0]																XXXXXXXXXXXXXXXX _B
001018 _H (ch3)																		
001020 _H (ch4)																		
Address	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Initial value
001004 _H (ch0)	bit	DMADA[31:16]																XXXXXXXXXXXXXXXX
00100C _H (ch1)																		
001014 _H (ch2)		DMADA[15:0]																XXXXXXXXXXXXXXXX _B
00101C _H (ch3)																		
001024 _H (ch4)																		

Detailed Bit of Transfer Source/Transfer Destination Address Setting Register (DMASA0 to 4/DMADA0 to 4)

The following describes the functions of the bits of each transfer source/transfer destination address setting register (DMASA0 to 4/DMADA0 to 4).

[Bits 31 to 0] DMASA (DMA Source Addr)*: Transfer source address setting

These bits set the transfer source address.

[Bits 31 to 0] DMADA (DMA Destination Addr)*: Transfer destination address setting

These bits set the transfer destination address.

If DMA transfer is activated, data in this register is stored in the counter buffer of the DMA-dedicated address counter and then the address is calculated according to the settings for the transfer operation. When the DMA transfer is completed, the contents of the counter buffer are written back to this register and then DMA ends. Thus, the address counter value during DMA operation cannot be read.

All registers have a dedicated reload register. When the register is used for a channel that is enabled for reloading of the transfer source/transfer destination address register, the initial value is automatically written back to the register when the transfer is completed. Other address registers are not affected.

- When reset: Not initialized.
- These bits are readable and writable. For this register, be sure to access these bits as 32-bit data.
- If these bits are read during transfer, the address before the transfer is read. If they are read after transfer, the next access address is read. Because the reload value cannot be read, it is not possible to read the transfer address in real time.

Note:

Do not set any of the DMAC's registers using this register. DMA transfer is not possible for the DMAC's registers themselves.

26.2.6 DMAC All-Channel Control Register (DMACR)

The DMAC all-channel control register (DMACR) controls the operation of the all five DMAC channels. Be sure to access this register using byte length.

This section describes the configuration and functions of the DMAC all-channel control register (DMACR).

■ Bit Configuration of DMAC All-Channel Control Register (DMACR)

Figure 26.2-5 "Bit Configuration of the DMAC All-Channel Control Register (DMACR)" shows the bit configuration of the DMAC all-channel control register (DMACR).

Figure 26.2-5 Bit Configuration of the DMAC All-Channel Control Register (DMACR)

Address	000240 _H	bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Initial value
			DMAE	-	-	PM01	DMAH[3:0]			-	-	-	-	-	-	-	-	-	
		bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	XXXXXXXXXXXXXXXX _B
			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

■ Detailed Bit of DMAC All-Channel Control Register (DMACR)

The following describes the bit functions of the DMAC all-channel control register (DMACR) bits.

[Bit 31] DMAE (DMA Enable): DMA operation enable

This bit controls the operation of all DMA channels.

If DMA operation is disabled with this bit, transfer operations on all channels are disabled regardless of the start/stop settings for each channel and the operating status. Any channel carrying out transfer cancels the requests and stops transfer at a block boundary. All start operations on each channel in a disabled state are disabled.

If this bit enables DMA operation, start/stop operations are enabled for each channel. Simply enabling DMA operation with this bit does not activate each channel.

MB91460 Series

DMA operation can be forced to stop by writing 0 to this bit. However, be sure to force stopping (0 write) only after temporarily stopping DMA using the DMAH[3:0] bits [Bit27 to 24 of DMACR]. If forced stopping is carried out without first temporarily stopping DMA, DMA stops, but the transfer data cannot be guaranteed. Check whether DMA is stopped using the DSS[2:0] bits [Bit18 to 16 of DMACB].

DMAE	Function
0	Disables DMA transfer on all channels. (initial value)
1	Enables DMA transfer on all channels.

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bit 28] PM01 (Priority mode ch0,ch1): Channel priority rotation

This bit is set to alternate priority for each transfer between Channel0 and Channel1.

PM01	Function
0	Fixes the priority. (ch0 > ch1)(initial value)
1	Alternates priority. (ch1 > ch0)

- When reset: Initialized to 0.
- This bit is readable and writable.

[Bits 27 to 24] DMAH (DMA Halt): DMA temporary stop

These bits control temporary stopping of all DMA channels. If these bits are set, DMA transfer is not performed on any channel before these bits are cleared.

When DMA transfer is activated after these bits are set, all channels remain temporarily stopped.

Transfer requests that occur on channels for which DMA transfer is enabled (DENB=1) while these bits are set are all enabled. The transfer can be started by clearing all these bits.

DMAH	Function
0000 _B	Enables the DMA operation on all channels. (initial value)
Other than 0000 _B	Temporarily stops DMA operation on all channels.

- When reset: Initialized to 0.
- These bits are readable and writable.

[Bits 30, 29, and 23 to 0] (Reserved): Unused bits

These bits are unused.

- A read value is undefined.

26.2.7 Other Functions

The MB91460 series has the DACKX, DEOP, DEOTX, and DREQ pins, which can be used for external transfer. These pins can also be used as general-purpose ports.

■ Pin Function of the DACKX, DEOP, DEOTX and DREQ pins

To use the DACKX, DEOP, DEOTX or DREQ pins for external transfer, a switch must be made from the port function to the DMA pin function.

To make the switch, set the PFR register.

MB91460 Series

26.3. DMA Controller (DMAC) Operation

A DMA controller (DMAC) is built into all FR family devices. The FR family DMAC is a multi-functional DMAC that controls data transfer at high speed without the use of CPU instructions.

This section describes the operation of the DMAC.

■ Principal Operations

- Functions can be set for each transfer channel independently.
- Once starting has been enabled, a channel starts transfer operation only after a specified transfer request has been detected.
- After a transfer request is detected, a DMA transfer request is output to the bus controller and the bus right is acquired by the bus controller before the transfer is started.
- The transfer is carried out as a sequence conforming to the mode settings made independently for the channel being used.

■ Transfer Mode

Each DMA channel performs transfer according to the transfer mode set by the MOD[1:0] bits of its DMACB register.

● Block/step transfer

Only a single block transfer unit is transferred in response to one transfer request. DMA then stops requesting the bus controller for transfer until the next transfer request is received.

The block transfer unit is the specified block size (BLK[3:0] of DMACA).

● Burst transfer

Transfer in response to one transfer request is carried out continuously for the number of times in the specified transfer count is reached.

The specified transfer count is the transfer count (BLK[3:0] of DMACA X DTC[15:0] of DMACA) X block size.

● Demand transfer

Transfer is carried out continuously until the transfer request input (detected with a level at the DREQ pin) from an external device or a specified transfer count is reached.

The specified transfer count in a demand transfer is the specified transfer count (DTC[15:0] of DMACA). The block size is always 1 and the register value is ignored.

■ Transfer Type

● 2-cycle transfer (normal transfer)

The DMA controller operates using a read operation and a write operation as its unit of operation.

Data is read from an address in the transfer source register and then written to another address in the transfer destination register.

● Fly-by transfer (external memory --> external I/O)

The DMA controller operates using a read operation as its unit of operation.

If DMA transfer is performed when fly-by transfer is set, DMA issues a fly-by transfer (read) request to the bus controller and the bus controller lets the external interface carry out the fly-by transfer (read).

- Fly-by transfer (external I/O --> external memory)

The DMA controller operates using a write operation as its unit of operation.

Otherwise, operation is the same as fly-by transfer (external memory --> external I/O) operation.

Access areas used for MB91460 series fly-by transfer must be external areas.

■ Transfer Address

The following types of addressing are available and can be set independently for each channel transfer source and transfer destination.

The method for specifying the address setting register (DMASA/DMADA) for a 2-cycle transfer and the method for a fly-by transfer are different.

- Specifying the address for a 2-cycle transfer

The value read from a register (DMASA/DMADA) in which an address has been set in advance is used as the address for access. After receiving a transfer request, DMA stores the address from the register in the temporary storage buffer and then starts transfer.

After each transfer (access) operation, the next access address is generated (increment/decrement/fixed selectable) by the address counter and then written to the temporary storage buffer. Because the contents of the temporary storage buffer are written back to the register (DMASA/DMADA) after each block transfer unit is completed, the address register (DMASA/DMADA) value is updated after each block transfer unit is completed, making it impossible to determine the address in real time during transfer.

- Specifying the address for a fly-by transfer

In a fly-by transfer, the value read from the transfer destination address register (DMADA) is used as the address for access. The transfer source address register (DMASA) is ignored. Be sure to specify an external area as the address to be set.

After receiving a transfer request, DMA stores the address from the register in the temporary storage buffer and then starts transfer.

After each transfer (access) operation, the next access address is generated (increment/decrement/fixed selectable) by the address counter and then written to the temporary storage buffer. Because the contents of this temporary storage buffer are written back to the register (DMADA) after each block transfer unit is completed, the address register (DMADA) value is updated after each block transfer unit is completed, making it impossible to determine the address in real time during transfer.

■ Transfer Count and Transfer End

- Transfer count

The transfer count register is decremented (-1) after each block transfer unit is completed. When the transfer count register becomes 0, counting for the specified transfer ends, and the transfer stops with the end code displayed or is reactivated *.

Like the address register, the transfer count register value is updated only after each block transfer unit.

*: If transfer count register reloading is disabled, the transfer ends. If reloading is enabled, the register value is initialized and then waits for transfer (DTCCR of DMACB)

- Transfer end

Listed below are the sources for transfer end. When transfer ends, a source is indicated as the end code (DSS[2:0] of DMACB).

MB91460 Series

- End of the specified transfer count (DMACA:BLK[3:0] x DMACA:DTC[15:0]) => Normal end
- A transfer stop request from a peripheral circuit or the external pin (DSTP) occurred => Error
- An address error occurred => Error
- A reset occurred => Reset

The transfer stop source is indicated (DSS) and the transfer end interrupt or error interrupt for the end source is generated.

26.3.1 Setting a Transfer Request

The following three types of transfer requests are provided to activate DMA transfer:

- External transfer request pin
- Built-in peripheral request
- Software request

Software requests can always be used regardless of the settings of other requests.

■ External Transfer Request Pin (DREQ 0-3)

A transfer request is generated by input to the input pin prepared for a channel.

The MB91460 series supports channels 0-3 (DREQ0-3).

If the input is valid at this point, the following sources are selected depending on the settings for the transfer mode and the start source:

● Edge detection

If the transfer mode is block, step, or burst transfer, select edge detection:

- Rising edge detection: Set with the transfer source selection register. When IS[4:0] of DMACA=01110_B.
- Falling edge detection: Set with the transfer source selection register. When IS[4:0] of DMACA=01111_B.

● Level detection

If the transfer mode is demand transfer, select level detection:

- H level detection: Set with the transfer source selection register. When IS[4:0] of DMACA=01110_B.
- L level detection: Set with the transfer source selection register. When IS[4:0] of DMACA=01111_B.

■ Built-in Peripheral Request

A transfer request is generated by an interrupt from the built-in peripheral circuit.

For each channel, set the peripheral's interrupt by which a transfer request is generated (When IS[4:0] of DMACA=1xxxx.)

The built-in peripheral request cannot be used together with an external transfer request.

Note:

Because an interrupt request used in a transfer request seems like an interrupt request to the CPU, disable interrupts from the interrupt controller (ICR register).

■ Software Request

A transfer request is generated by writing to the trigger bit of a register (STRG of DMACA).

The software request is independent of the external transfer request pin and built-in peripheral request and can

always be caused.

If a software request occurs together with a start (transfer enable) request, the transfer is started by immediate output of a DMA transfer request to the bus controller.

26.3.2 Transfer Sequence

The transfer type and the transfer mode that determine, for example, the operation sequence after DMA transfer has started can be set independently for each channel (Settings for TYPE[1:0] and MOD[1:0] of DMACB).

■ Selection of the Transfer Sequence

The following sequence can be selected with a register setting:

- Burst 2-cycle transfer
- Demand 2-cycle transfer
- Block/step 2-cycle transfer
- Burst fly-by transfer
- Demand fly-by transfer
- Block/step fly-by transfer

● Burst 2-cycle transfer

In a burst 2-cycle transfer, as many transfers as specified by the transfer count are performed continuously for one transfer source. For a 2-cycle transfer, all 32-bit areas can be specified using a transfer source/transfer destination address.

A peripheral transfer request, software transfer request, or external pin (DREQ) edge input detection request can be selected as the transfer source.

Table 26.3-1 Specifiable transfer addresses (burst 2-cycle transfer)

Transfer source addressing	Direction	Transfer destination addressing
All 32-bit areas specifiable	=>	All 32-bit areas specifiable

The following are some features of a burst transfer:

When one transfer request is received, transfer is performed continuously until the transfer count register reaches 0.

The transfer count is the transfer count x block size (BLK[3:0] of DMACA x DTC[15:0] of DMACA).

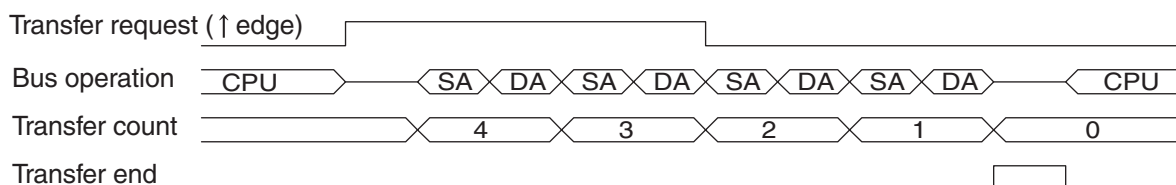
Another request occurring during transfer is ignored.

If the reload function of the transfer count register is enabled, the next request is accepted after transfer ends.

If a transfer request for another channel with a higher priority is received during transfer, the channel is switched at the boundary of the block transfer unit. Processing resumes only after the transfer request for the other channel is cleared.

Figure 26.3-1 shows an example of burst transfer for a start on an external pin rising edge, number of blocks = 1, and transfer count = 4:

Figure 26.3-1 Example of burst transfer for a start on an external pin rising edge



● Burst fly-by transfer

A burst fly-by transfer has the same features as a 2-cycle transfer except that the transfer area can only be external areas, and the transfer unit is read (external memory --> external I/O) or write (external I/O --> external memory) only.

Table 26.3-2 Specifiable transfer addresses (burst fly-by transfer)

Transfer source addressing	Direction	Transfer destination addressing
Specification not required (invalid)	None	External area

● Demand Transfer 2-Cycle Transfer

A demand transfer sequence is generated only if H level or L level of an external pin is selected as a transfer request. Select the level with IS[3:0] of DMACA.

The following are some features of a continuous transfer:

- Each transfer operation of a transfer request is checked. While the external input level is within the range of the specified transfer request levels, transfer is performed continuously without the request being cleared. If the external input changes, the request is cleared and the transfer stops at the transfer boundary. This operation is repeated for the number of times specified by the transfer count.
- Otherwise, operations are the same as those of a burst transfer.

Figure 26.3-2 shows an example of demand transfer for a start with the external pin at H level, number of blocks = 1, and transfer count = 3:

Figure 26.3-2 Example of demand transfer for a start with the external pin at H level

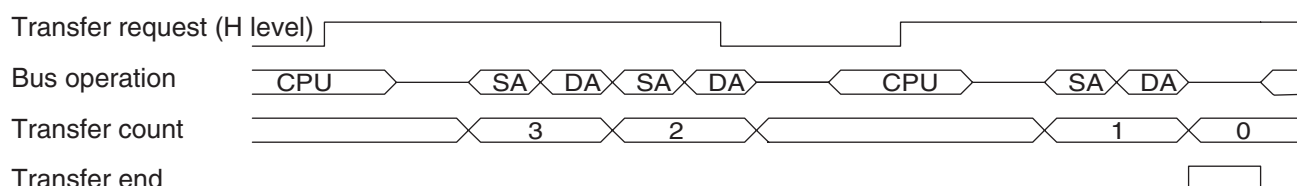


Table 26.3-3 Specifiable transfer addresses (demand transfer 2-cycle transfer)

Transfer source address	Direction	Transfer destination addressing
External area	=>	External area
External area	=>	Built-in IO
External area	=>	Built-in RAM
Built-in IO	=>	External area
Built-in RAM	=>	External area

Note:

For a demand transfer, be sure to set an external area address for the transfer source or transfer destination or both. Since DMA transfer is adjusted to the external bus timing in demand transfer mode, access to external areas is always needed.

● Demand transfer fly-by transfer

A demand transfer fly-by transfer has the same features as a 2-cycle transfer except that the transfer area can only be external areas, and the transfer unit is read (external memory --> external I/O) or write (external I/O --> external memory) only.

Table 3.2-3 Specifiable transfer addresses (demand transfer fly-by transfer)

Transfer source addressing	Direction	Transfer destination addressing
Specification not required (invalid)	=>	External area

● Step/block transfer 2-cycle transfer

For a step/block transfer (Transfer for each transfer request is performed as many times as the specified block count), all 32-bit areas can be specified as the transfer source/transfer destination address.

Table 26.3-4 Specifiable transfer addresses (step/block transfer 2-cycle transfer)

Transfer source addressing	Direction	Transfer destination addressing
All 32-bit areas specifiable	=>	All 32-bit areas specifiable

[Step transfer]

If 1 is set as the block size, a step transfer sequence is generated.

The following are some features of a step transfer:

- If a transfer request is received, the transfer request is cleared after one transfer operation and then the transfer is stopped (The DMA transfer request to the bus controller is canceled).
- Another request occurring during transfer is ignored.

MB91460 Series

- If a transfer request for another channel with a higher priority is received during transfer, the channel is switched after the running transfer is stopped. After finishing the higher priority transfer the stopped transfer is restarted. Priority in a step transfer is valid only if transfer requests occur simultaneously.

[Block transfer]

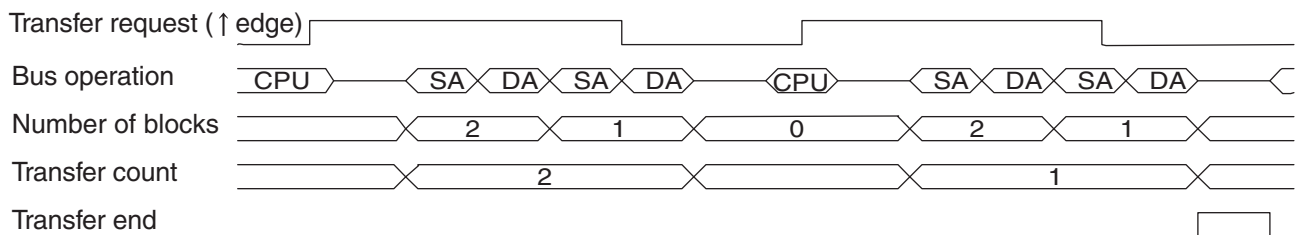
If any value other than 1 is specified as the block size, a block transfer sequence is generated.

The following are some features of a block transfer:

- The block transfer has the same features as those of a step transfer except that one transfer unit consists of multiple transfer cycle counts (number of blocks).

Figure 26.3-4 shows an example of block transfer for a start for an external pin on a rising edge, number of blocks = 2, and transfer count = 2

Figure 26.3-4 Example of block transfer for a start for an external pin on a rising edge



● Step/block transfer fly-by transfer

This transfer has the same features as those of a 2-cycle transfer except that the transfer area can only be external areas, and the transfer unit is read (external memory --> external I/O) or write (external I/O --> external memory) only.

Table 26.3-5 Specifiable transfer addresses (step/block transfer fly-by transfer)

Transfer source addressing	Direction	Transfer destination addressing
Specification not required (ignored)	None	External area

26.3.3 General Aspects of DMA Transfer

This section describes the block size for DMA transfers and the reload operation.

■ Block Size

- The unit and increment for transfer data is a set of (the number set in the block size specification register x data width) data.
- Since the amount of data transferred in one transfer cycle is determined by the value specified as the data width, one transfer unit consists of the number of transfer cycles for the specified block size.
- If a transfer request with a higher priority is received during transfer or if a temporary stop request for a transfer occurs, the transfer stops only at the transfer unit boundary, whether or not the transfer is a block transfer. This arrangement makes it possible to protect data for which division or temporary stopping is not desirable. However, if the block size is large, response time decreases.
- Transfer stops immediately only when a reset occurs, in which case the data being transferred cannot be guaranteed.

■ Reload Operation

In this module, the following three types of reloading can be set for each channel:

● Transfer count register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer count register again and waiting for a start request starts.

Set this type of reloading when the entire transfer sequence is to be performed repeatedly.

If reload is not specified, the count register value remains 0 after the transfer is performed the specified number of times and no further transfer is performed.

● Transfer source address register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer source address register again.

Set this type of reloading when transfer is to be repeated from a fixed area in the transfer source address area.

If reload is not specified, the transfer source address register value after the transfer is performed the specified number of times becomes the next address. Use this type when the address area is not fixed.

● Transfer destination address register reloading

After transfer is performed the specified number of times, the initial value is set in the transfer destination address register again.

Set this type of reloading when transfer is to be repeated to a fixed area in the transfer destination address area.

(The processing hereafter is the same as described in "Transfer source address register reloading" above.)

- If only reloading of the transfer source/transfer destination register is enabled, restart after transfer is performed the specified number of times is not implemented and only the values of each address register are set.

● Special examples of operating mode and the reload operation

- If transfer is performed in continuous transfer mode by external pin input level detection and transfer count register reloading is used, transfer continues by reloading even though transfer ends during continuous input. Also in this case, an end code is set.
- If it is preferable that processing stops when data transfer ends and starts after input is detected again, do not specify reload.
- For a transfer in burst, block, or step transfer mode, transfer stops temporarily after reload when data transfer ends. Transfer does not start until new transfer request input is detected.

26.3.4 Addressing Mode

Specify the transfer destination/transfer source address independently for each transfer channel.

■ Address Register Specifications

The following two methods are provided to specify an address register. The method specified depends on the transfer sequence.

- In 2-cycle transfer type, set the transfer source address in the transfer source address setting register (DMASA) and the transfer destination address in the transfer destination address setting register (DMADA).
- In fly-by transfer type, specify the external memory address in the transfer destination address setting register

(DMADA). In this case, the value in the transfer source address setting register (DMASA) is ignored.

■ Features of the Address Register

This register has the maximum 32-bit length. With 32-bit length, all space in the memory map can be accessed.

■ Function of the Address Register

- The address register is read in each access operation and the read value is sent to the address bus.
- At the same time, the address for the next access is calculated by the address counter and the address register is updated using the calculated address.
- For address calculation, increment or decrement is selected independently for each channel, transfer destination, and transfer source. The address increment/decrement width is specified by the address count size register (SASZ/DASZ of DMACB).
- If reloading is not enabled, the address resulting from the address calculation of the last address remains in the address register when the transfer ends.
- If reloading is enabled, the initial value of the address is reloaded.

Notes:

- If an overflow or underflow occurs as a result of 32-bit length full address calculation, an address error is detected and transfer on the relevant channel is stopped. Refer to the description for the items related to the end code.
- Do not set any of the DMAC's registers as the address register.
- For demand transfer, be sure to set an address in an external area for the transfer source, transfer destination, or both.
- Do not let the DMAC transfer data to any of the DMAC's registers.

26.3.5 Data Types

Select the data width transferred in one transfer operation from the following:

- Byte
- Halfword
- Word

■ Data Length (Data width)

Since the word boundary specification is also observed in DMA transfer, different low-order bits are ignored if an address with a different data length is specified for the transfer destination/transfer source address.

- Byte: The actual access address and the addressing match.
- Halfword: The actual access address has 2-byte length starting with 0 as the lowest-order bit.
- Word: The actual access address has a 4-byte length starting with 00 as the lowest-order 2 bits.

If the lowest-order bits in the transfer source address and transfer destination address are different, the addresses as set are output on the internal address bus. However, each transfer target on the bus is accessed after the addresses are corrected according to the above rules.

26.3.6 Transfer Count Control

Specify the transfer count within the range of the maximum 16-bit length (1 to 65536).

■ Transfer Count Control

Set the transfer count value in the transfer count register (DTC of DMACA).

The register value is stored in the temporary storage buffer when the transfer starts and is decremented by the transfer counter. When the counter value becomes 0, end of transfer end for the specified count is detected, and the transfer on the channel is stopped or waiting for a restart request starts (when reload is specified).

The following are some features of the group of transfer count registers:

- Each register has 16-bit length.
- All registers have a dedicated reload register.
- If transfer is activated when the register value is 0, transfer is performed 65536 times.

■ Reload Operation

- The reload operation can be used only if reloading is enabled in a register that allows reloading.
- When transfer is activated, the initial value of the count register is saved in the reload register.
- If the transfer counter counts down to 0, end of transfer is reported and the initial value is read from the reload register and written to the count register.

26.3.7 CPU Control

When a DMA transfer request is accepted, DMA issues a transfer request to the bus controller.

The bus controller passes the right to use the internal bus to DMA at a break in bus operation and DMA transfer starts.

■ DMA Transfer and Interrupts

- During DMA transfer, interrupts are generally not accepted until the transfer ends.
- If a DMA transfer request occurs during interrupt processing, the transfer request is accepted and interrupt processing is stopped until the transfer is completed.
- If, as an exception, an NMI request or an interrupt request with a higher level than the hold suppress level set by the interrupt controller occurs, DMAC temporarily cancels the transfer request via the bus controller at a transfer unit boundary (one block) to temporarily stop the transfer until the interrupt request is cleared. In the meantime, the transfer request is retained internally. After the interrupt request is cleared, DMAC reissues a transfer request to the bus controller to acquire the right to use the bus and then restarts DMA transfer.

■ Suppressing DMA

When an interrupt source with a higher priority occurs during DMA transfer, an FR family device interrupts the DMA transfer and branches to the relevant interrupt routine. This feature is valid as long as there are any interrupt requests. When all interrupt sources are cleared, the suppression feature no longer works and the DMA transfer is restarted by the interrupt processing routine. Thus, if the restart of DMA transfer should be suppressed after clearing interrupt sources in the interrupt source processing routine at a level that interrupts DMA transfer, the DMA suppress function should be used. The DMA suppress function can be activated by writing any value other than 0 to the DMAH[3:0] bits of the DMA all-channel control register and can be stopped by writing 0 to these bits.

This function is mainly used in the interrupt processing routines. Before the interrupt sources in an interrupt processing routine are cleared, the DMA suppress register is incremented by 1. If this is done, then no DMA transfer is performed. After interrupt processing, decrement the DMAH[3:0] bits by 1 before returning. If multiple interrupts have occurred, DMA transfer continues to be suppressed since the DMAH[3:0] bits are not 0 yet. If a single interrupt has occurred, the DMAH[3:0] bits become 0. DMA requests are then enabled immediately.

MB91460 Series

Note:

- Since the register has only four bits, this function cannot be used for multiple interrupts exceeding 15 levels.
- Be sure to assign the priority of the DMA tasks at a level that is at least 15 levels higher than other interrupt levels.

26.3.8 External Bus Request Arbitration (BRQ)

When a device is operating in external bus extended mode, an external bus request (BRQ) function can be used. The relationship between external bus requests and DMA transfer requests by this module when the external bus request function can be used is described below.

■ DMA Transfer Request during External Hold

DMA transfer is started when an external bus area is accessed, DMA transfer is temporarily stopped. When the external hold is released, DMA transfer is restarted.

■ External Hold Request During DMA Transfer

The device is externally held. When an external bus area is accessed by DMA transfer, DMA transfer is temporarily stopped. When the external hold is released, DMA transfer is restarted.

■ Simultaneous Occurrence of a DMA Transfer Request and an External Hold Request

The device is externally held and internal DMA transfer is started. When an external bus area is accessed by DMA transfer, DMA transfer is temporarily stopped. When the external hold is released, DMA transfer is restarted.

26.3.9 Operation from Starting to End/Stopping

Starting of DMA transfer is controlled independently for each channel, but before transfer starts, the operation of all channels needs to be enabled. This section describes operation from starting to end/stopping.

■ Operation Start

● Enabling operation for all channels

Before activating each DMAC channel, operation for all channels needs to be enabled in advance with the DMA operation enable bit (DMAE of DMACR). All start settings and transfer requests that occurred before operation is enabled are invalid.

● Starting transfer

The transfer operation can be started by the operation enable bit of the control register for each channel. If a transfer request to an activated channel is accepted, the DMA transfer operation is started in the specified mode.

● Starting from a temporary stop

If a temporary stop occurs before starting with channel-by-channel or all-channel control, the temporary stopped state is maintained even though the transfer operation is started. If transfer requests occur in the meantime, they are accepted and retained. When temporary stopping is released, transfer is started.

■ Transfer Request Acceptance and Transfer

Sampling for transfer requests set for each channel starts after starting.

If edge detection is selected for the external pin start source and a transfer request is detected, the request is retained within DMAC until the clear conditions are met (when the external pin start source is selected for block, step, or burst transfer).

If level detection or peripheral interrupt start is selected for the external pin start source, DMAC continues the transfer until all transfer requests are cleared. When they are cleared, DMAC stops the transfer after one transfer unit (demand transfer or peripheral interrupt start).

Since peripheral interrupts are handled as level detection, use interrupt clear by DMA to handle the interrupts.

Transfer requests are always accepted while other channel requests are being accepted and transfer performed. The channel that will be used for transfer is determined for each transfer unit after priority has been checked.

■ Clearing Peripheral Interrupts by DMA

This DMA has a function that clears peripheral interrupts. This function works when peripheral interrupt is selected as the DMA start source (when IS[4:0]=1xxxx_B).

Peripheral interrupts are cleared only for the set start sources. That is, only the peripheral functions set by IS[4:0] are cleared.

The timing for clearing an interrupt depends on the transfer mode (See Section 26.4."Operation Flowcharts").

- Block/step transfer: If block transfer is selected, a clear signal is generated after one block (step) transfer.
- Burst transfer: If burst transfer is selected, a clear signal is generated after transfer is performed the specified number of times.
- Demand transfer: Since only start requests from external pins are supported in demand transfer, no clear signal is generated.

■ Temporary Stopping

DMA transfer is stopped temporary in the following cases:

- Setting of temporary stopping by writing to the control register (Set independently for each channel or all channels simultaneously)

If temporary stopping is set using the temporary stop bit, transfer on the corresponding channel is stopped until release of temporary stopping is set again. The DSS bits for temporary stopping can be checked.

- NMI/hold suppress level interrupt processing

If an NMI request or an interrupt request with a higher level than the hold suppress level occurs, all channels on which transfer is in progress are temporarily stopped at the boundary of the transfer unit and the bus right is returned to give priority to NMI/interrupt processing. Transfer request accepted during NMI/interrupt processing are retained, initiating a wait for completion of NMI processing.

Channels for which requests are retained restart transfer after NMI/interrupt processing is completed.

■ Operation End/Stopping

The end of DMA transfer is controlled independently for each channel. It is also possible to disable operation for all channels at once.

- Transfer end

If reloading is disabled, transfer is stopped, "Normal end" is displayed as the end code, and all transfer requests are disabled after the transfer count register becomes 0 (Clear the DENB bit of DMACA).

If reloading is enabled, the initial value is reloaded, "Normal end" is displayed as the end code, and a wait for transfer requests starts after the transfer count register becomes 0 (Do not clear the DENB bit of DMACA).

MB91460 Series

● Disabling all channels

If the operation of all channels is disabled with the DMA operation enable bit DMAE, all DMAC operations, including operations on active channels, are stopped. Then, even if the operation of all channels is enabled again, no transfer is performed unless a channel is restarted. In this case, no interrupt whatever occurs.

■ Stopping Due To an Error

In addition to normal end after transfer for the number of times specified, stopping as the result of various types of errors and the forced stopping are provided. (See "[Bits 18 to 16] DSS2 to 0 (DMA Stop Status)*: Transfer stop source indication" on P. 484)

● Transfer stop requests from peripheral circuits

Depending on the peripheral circuit that outputs a transfer request, a transfer stop request is issued when an error is detected (Example: Error when data is received at or sent from a communications system peripheral).

The DMAC, when it receives such a transfer stop request, displays "Transfer stop request" as the end code and stops the transfer on the corresponding channel.

Table 26.3-6 Stopping due to an Error

IS	EIS	Function	Transfer stop request
10110	0000	USART 0 RX *1	Yes
11000	0000	USART 1 RX *1	Yes
11010	0000	USART 4 RX *1	Yes
11100	0000	USART 5 RX *1	Yes
10000	0011	USART 0 RX *1	Yes
10010	0011	USART 1 RX *1	Yes
10100	0011	USART 2 RX *1	Yes
10110	0011	USART 3 RX *1	Yes
11000	0011	USART 4 RX *1	Yes
11010	0011	USART 5 RX *1	Yes
11100	0011	USART 6 RX *1	Yes
11110	0011	USART 7 RX *1	Yes
10000	0100	USART 8 RX *1	Yes
10010	0100	USART 9 RX *1	Yes
10100	0100	USART 10 RX *1	Yes
10110	0100	USART 11 RX *1	Yes
11000	0100	USART 12 RX *1	Yes
11010	0100	USART 13 RX *1	Yes
11100	0100	USART 14 RX *1	Yes
11110	0100	USART 15 RX *1	Yes
others	others		None

*1 : A transfer stop request is issued when an error is detected

For details of the conditions under which a transfer stop request is generated, see the specifications for each peripheral circuit.

■ Occurrence of an Address Error

If inappropriate addressing, as shown below in parenthesis, occurs in an addressing mode, an address error is detected (if an overflow or underflow occurs in the address counter when a 32-bit address is specified).

If an address error is detected, "An address error occurred" is displayed as the end code and transfer on the corresponding channel is stopped. (See "[Bits 18 to 16] DSS2 to 0 (DMA Stop Status)*: Transfer stop source indication" on P. 484)

26.3.10 DMAC Interrupt Control

Independent of peripheral interrupts that become transfer requests, interrupts can also be output for each DMAC channel.

■ DMAC Interrupt Control

The following interrupts can be output for each DMAC channel:

- Transfer end interrupt: Occurs only when operation ends normally.
- Error interrupt: Transfer stop request due to a peripheral circuit (error due to a peripheral)
- Error interrupt: Occurrence of address error (error due to software)

All of these interrupts are output according to the meaning of the end code.

An interrupt request can be cleared by writing 000_B to DSS2 to 0 (end code) of DMACB. Be sure to clear the end code by writing 000_B before restarting.

If reloading is enabled, the transfer is automatically restarted. At this point, however, the end code is not cleared and is retained until a new end code is written when the next transfer ends.

Since only one end source can be displayed in an end code, the result after considering the order of priority is displayed when multiple sources occur simultaneously. The interrupt that occurs at this point conforms to the displayed end code.

The following shows the priority for displaying end codes (in order of decreasing priority):

- Reset
- Clearing by writing 000_B
- Peripheral stop request or external pin input (DSTP) stop request
- Normal end
- Stopping when address error detected
- Channel selection and control

■ DMA Transfer during SLEEP

- The DMAC can also operate in SLEEP state.
- If there are operations during SLEEP state anticipated, the following should be noticed:
 - Since the CPU is stopped, DMAC registers cannot be rewritten. Make settings before SLEEP state is entered.
 - The SLEEP state is released by an interrupt. Thus, if a peripheral interrupt is selected as the DMAC start source, interrupts must be disabled by the interrupt controller.

- If the SLEEP state should not be released with a DMAC end interrupt, disable these interrupts.

26.3.11 Channel Selection and Control

Up to five channels can be simultaneously set as transfer channels. In general, an independent function can be set for each channel.

■ Priority Among Channels

Since DMA transfer is possible only on one channel at a time, priority must be set for the channels.

Two modes, fixed and rotation, are provided as the priority settings and can be selected for each channel group (described later).

● Fixed mode

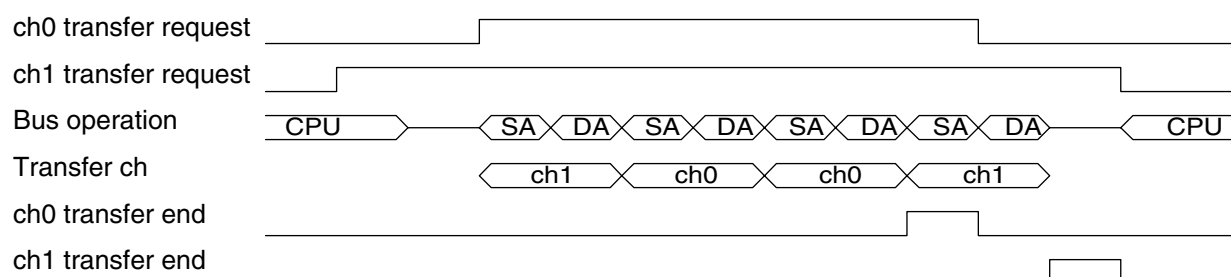
The order of priority is fixed by channel number, with priority decreasing from channel 0 to channel 4:

(ch.0 > ch.1 > ch.2 > ch.3 > ch.4)

If a transfer request with a higher priority is received during a transfer, the transfer channel becomes the channel with the higher priority when the transfer for the transfer unit (number set in the block size specification register x data width) ends.

When higher priority transfer is completed, transfer is restarted on the previous channel.

Figure 26.3-5 Timing Example in Fixed Mode

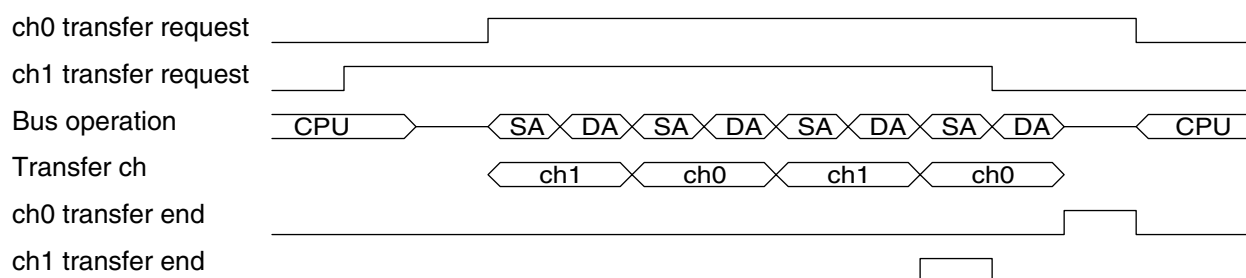


● Rotation mode (ch.0 to ch.1 only)

When operation is enabled, the initial states have the same order that they would have in fixed mode, but at the end of each transfer operation, the priority of the channels is reversed. Thus, if more than one transfer request is output at the same time, the channel is switched after each transfer unit.

This mode is effective when continuous or burst transfer is set.

Figure 26.3-6 Timing Example in Rotation Mode



■ Channel Group

The order of priority is set as shown in the following table.

MODE	Priority	Remarks
Fixed	ch0 > ch1	—
Rotation	ch0 > ch1 ↑ ↓ ch0 < ch1	The initial state is the top row. If transfer occurs for the top row, the priority is reversed.

26.3.12 Supplement on External Pin and Internal Operation Timing

This section provides supplementary information about external pins and internal operation timing.

■ Minimum Effective Pulse Width of the DREQ pin Input

Only channels 0-3 are applicable for the MB91460 series.

In all transfer modes for burst, step, block, and demand transfers, the minimum width required is five system clock cycles (5 cycles of CLK_T).

Note:

DACKX output does not indicate acceptance of DREQ input. DREQ input is always accepted if DMA is enabled but transfer has not started. Therefore, it is not necessary to retain DREQ input until DACKX output is asserted (except in demand transfer mode).

■ Negate Timing of the DREQ Pin Input when a Demand Transfer Request is Stopped

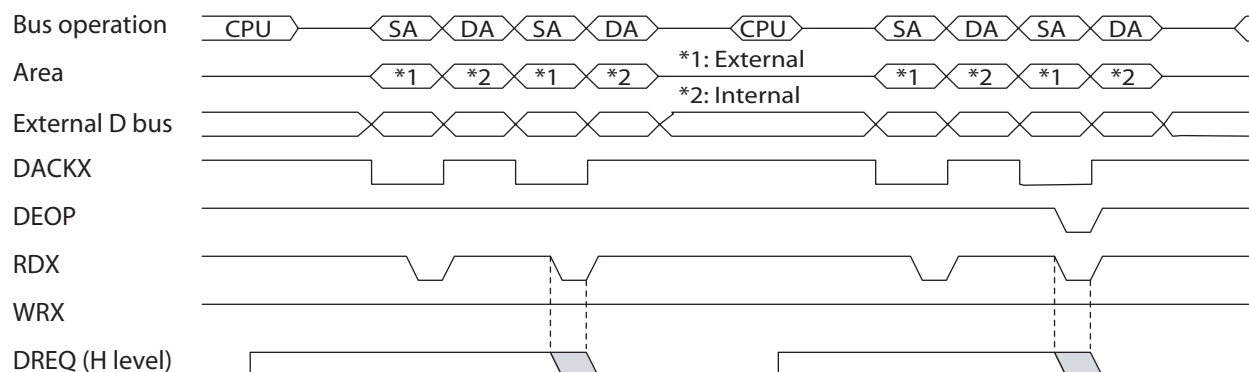
● For 2-cycle transfer

For a demand transfer, be sure to set an address in an external area for the transfer source, the transfer destination, or both.

- If the transfer type is external <--> external: Negate before the last sense timing of the clock in the L section of the external WRX_n pin output when accessing the transfer source for the last DMA transfer (section where DACKX = L and WRX_n = L). If DREQ is negated later than this, a DMA request may be sensed, resulting in negation until the next transfer.
- If the transfer type is external <--> internal: Negate before the last sense timing of the clock in the L section of the external RDX pin output when accessing the transfer source for the last DMA transfer (Section where DACKX = L and RDX = L). If DREQ is negated later than this, a DMA request may be sensed, resulting in negation until the next transfer

Figure 26.3-7 shows a negate timing example of the DREQ pin input for 2-cycle external transfer --> internal transfer:

Figure 26.3-7 Negate timing example of the DREQ pin input for 2-cycle external --> internal



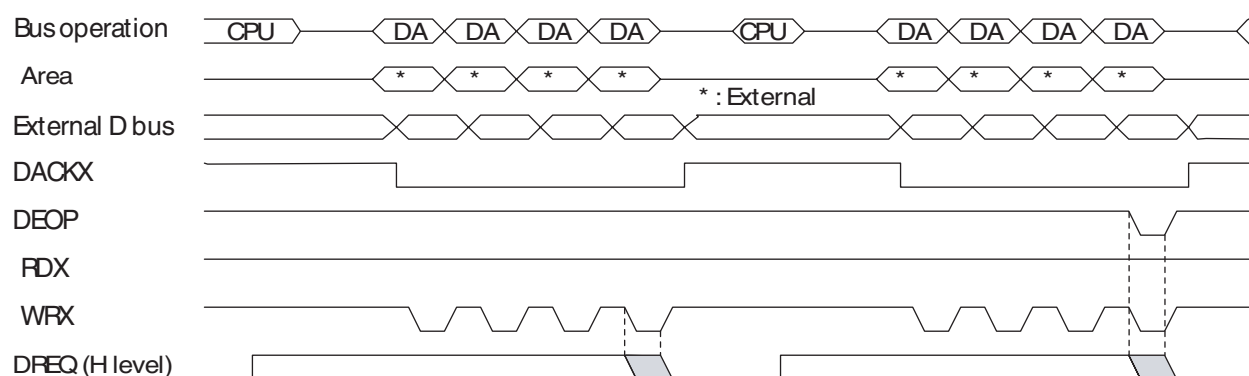
- If the transfer is internal <--> external: Negate before the last sense timing of the clock in the L section of the external WRXn pin output when accessing the transfer source for the last DMA transfer (Section of DACKX = 1 and WRXn = L). If DREQ is negated later than this, a DMA request may be sensed, resulting in negation until the next transfer.

● For fly-by (read/write) transfer

For a demand transfer, be sure to set an address in an external area for the transfer destination.

- For fly-by (read) transfer: After the IOWRX pin output for the last DMA transfer goes to the H level, negate DREQ while the external RDX pin output is at the L level. (section where DACKX=L & RDX=L). If DREQ is negated later than this, the negation may continue until the next transfer.
- For fly-by (write) transfer: After the external WRXn pin output for the last DMA transfer goes to the H level, negate DREQ while IORDX is at the L level. (section where DACKX=L & RDX=L). If DREQ is negated later than this, the negation may continue until the next transfer.

Figure 26.3-8 Negate timing example of the DREQ pin input for fly-by (write) transfer



■ **Input Timing of the DREQ Pin for Continuing Transfer over the Same Channel**

- For burst, step, block, and demand transfers

Operation in which transfer is continued over the same channel by the DREQ pin input cannot be guaranteed. If DREQ is reasserted at the fastest timing to clear requests retained internally after the transfer ends, at least one system clock cycle (one CLK output cycle) is provided to detect transfer requests for other channels. If, as a result, a transfer request for another channel with a higher priority is detected, transfer on that channel will be started.

Even if DREQ is reasserted earlier, it is ignored because the transfer has not been completed. If no transfer requests for other channels occur, transfer over the same channel is restarted by reasserting DREQ when the DACKX pin output is asserted.

■ Output Timing of DACKX Pin

The DACKX output of this DMAC indicates that transfer with respect to an accepted transfer request is being performed.

The output of DACKX is basically synchronized with the address output of external bus access timing. To use DACKX output, it is necessary to enable the DACKX output with a port.

■ Output Timing of the DEOP Pin

- The DEOP output of this DMA indicates that DMA transfer for the specified number of times of the accepted channel has been completed.
- DEOP output is output when access to an external area of the last transfer block starts. Thus, if any value other than 1 is set (block transfer mode) as the block size, DEOP is output when the last data of the last block is transferred. In this case, the acceptance of the next DREQ is already started even during transfer (before DEOP output) if the DACKX pin output is asserted.
- The DEOP output is synchronized with RDX and WRXn of external bus access timing. However, if the transfer source/transfer destination is internal access, DEOP is not output. To use DEOP output, it is necessary to enable the DEOP output using the port register.
-

■ Input Timing of the DEOTX Pin

- In all transfer modes of burst, step, block, and demand transfers, a width of at least 5 system clock cycles ($=1/2\phi$, 5 cycles of the DMA operating clock CLKB) are needed.
- As with DREQ, it is recommended to use DEOP input timing in synchronization with external access (Use the DACKX output and the signal decoded by RD or WR).
- Use the pin input to force DMA transfer to stop. Although transfer can be forced to stop by using this pin input, the status register (DSS[2:0] of DMACB) indicates "Transfer stop request" and is handled as an error. If interrupts are enabled, interrupts will occur.
- Since this function is shared with the DEOP pin, both functions cannot be used. Set switching of functions with the port register.

■ If an External Pin Transfer Request is Reentered During Transfer

- For burst, step, and block transfers

While the DACKX signal is asserted within the DMAC, the next transfer request, if it is entered, is disabled. However, since operation of the external bus control unit and operation of the DMAC are not completely synchronous, the circuit must be initialized to create DREQ pin input using DACKX and DEOP output to enable transfer requests by using DREQ input.

- For a demand transfer

If reloading of the transfer count register is specified when transfer for as many transfers as specified has been completed, another transfer request is accepted.

■ If Another Transfer Request Occurs During Block Transfer

No request is detected before the transfer of the specified blocks is completed. At the block boundaries, transfer requests accepted at that time are evaluated and then transfer on the channel with the highest priority is performed.

■ Transfer Between External I/O and External Memory

As targets of transfer by the DMAC, external I/O and external memory are not distinguished. Specify an external I/O as a fixed external address.

To perform fly-by transfer, set the address of external memory in the transfer destination address register. For external I/O, use DACKX output and the signal decoded by the read signal RDX or write signal WRXn pin.

■ AC Characteristics of DMAC

DREQ pin input, DEOTX pin input, DACKX pin output, and DEOP pin output are provided as the external pins related to the DMAC. Output timing is synchronized with external bus access (refer to the AC standard for the DMAC).

26.4. Operation Flowcharts

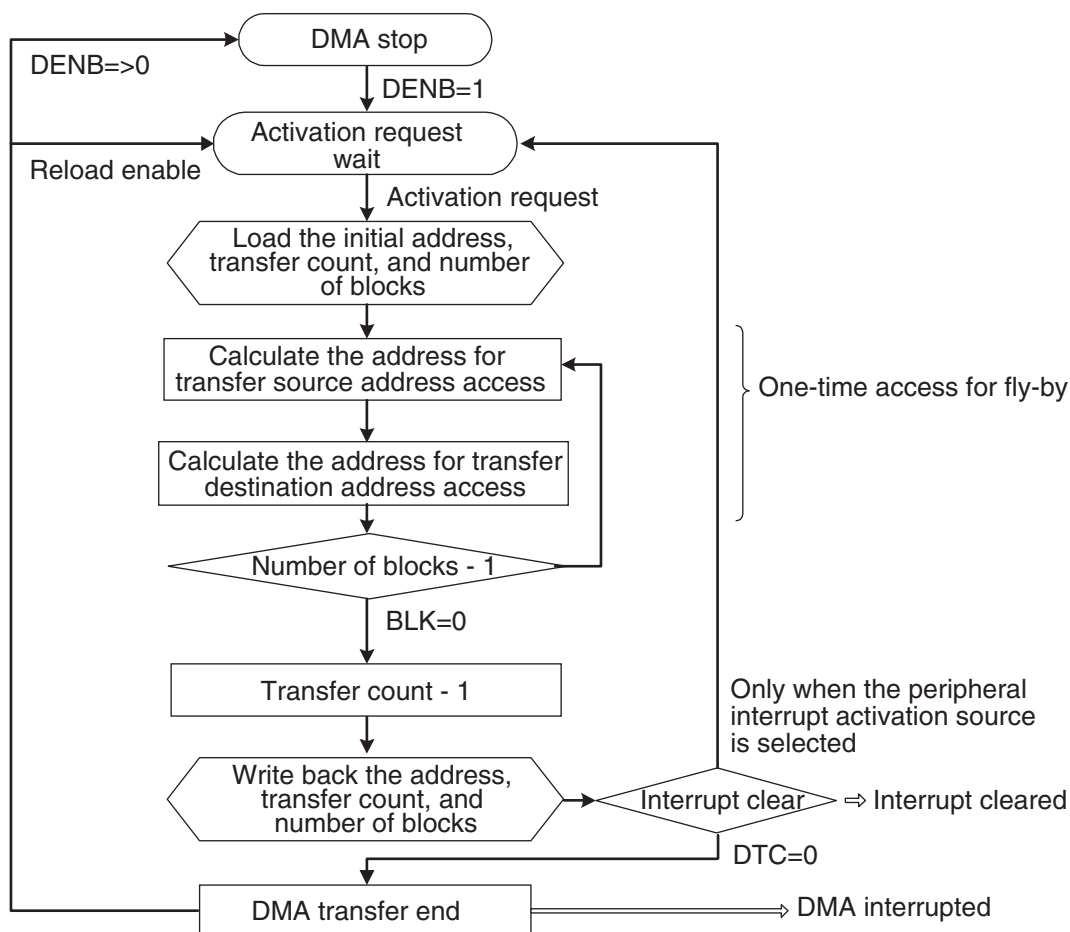
This section contains operation flowcharts for the following transfer modes:

- Block transfer
- Burst transfer
- Demand transfer

■ Block Transfer

Figure 26.4-1 "Operation Flowchart for Block Transfer" shows the flowchart for block transfer.

Figure 26.4-1 Operation Flowchart for Block Transfer



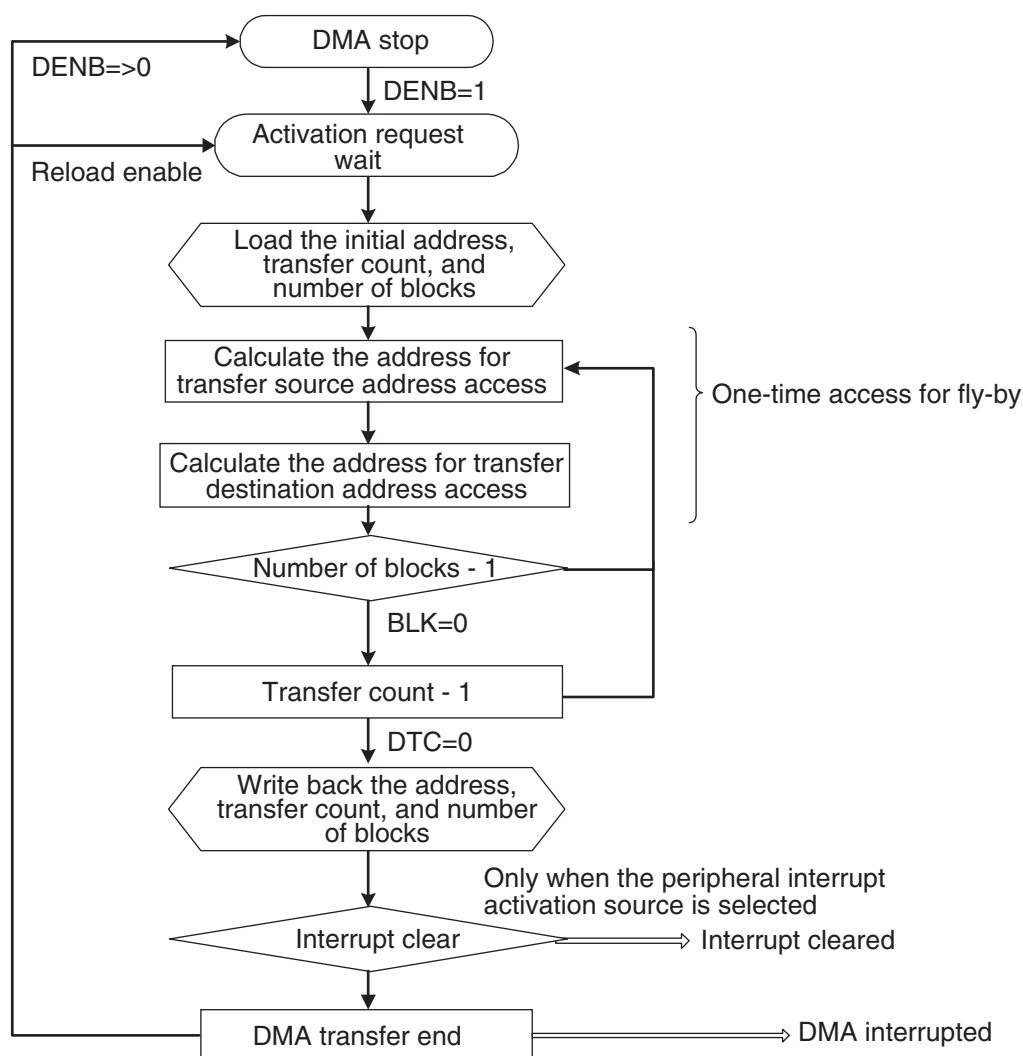
Block transfer

- Can be activated by all activation sources (selection).
- Can access to all areas.
- The number of blocks can be set.
- Interrupt clear is issued when transfer of the specified number of blocks is completed.
- The DMA interrupt is issued when transfer for the number of times specified is completed.

■ Burst Transfer

Figure 26.4-2 "Operation Flowchart for Burst Transfer" shows the operation flowchart for burst transfer.

Figure 26.4-2 Operation Flowchart for Burst Transfer

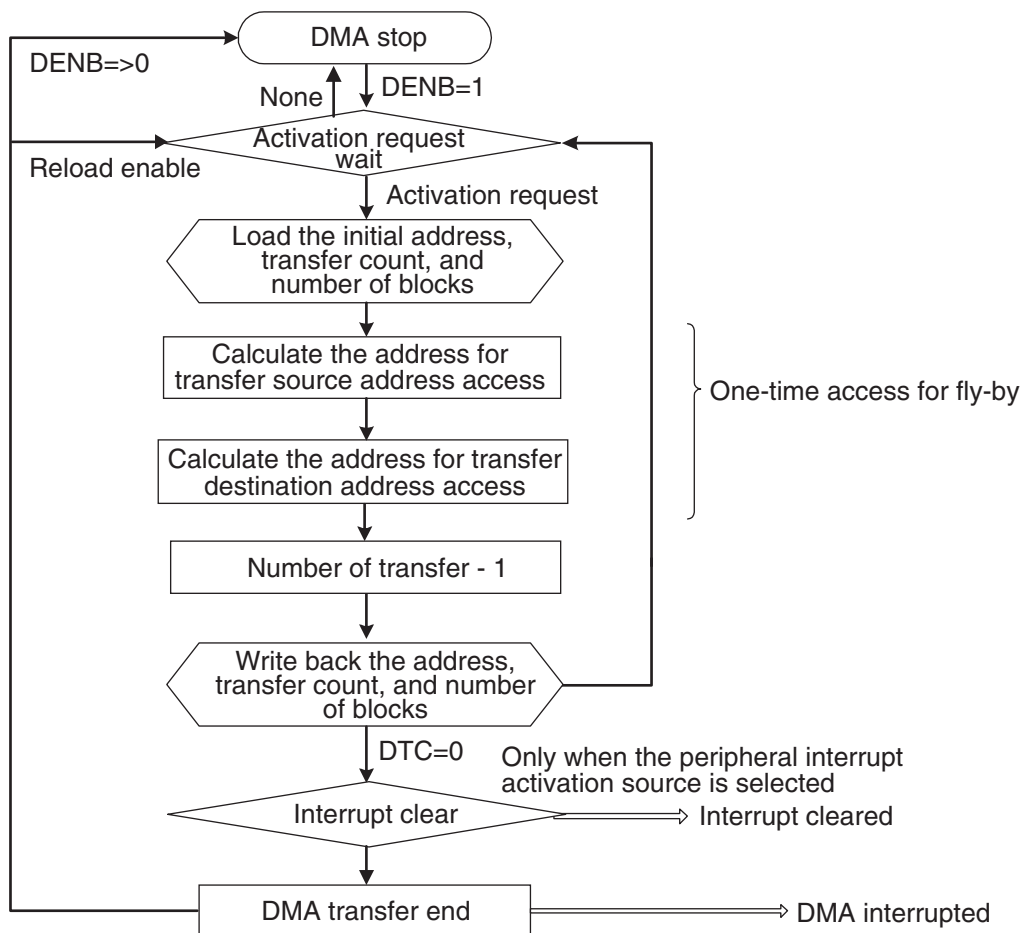
**Burst transfer**

- Can be activated by all activation sources (selection).
- Can access to all areas.
- The number of blocks can be set.
- Interrupt clear and the DMA interrupt are issued when transfer for the number of times specified is completed.

■ Demand Transfer

Figure 26.4-3 "Operation Flowchart for Demand Transfer" shows the operation flowchart for demand transfer.

Figure 26.4-3 Operation Flowchart for Demand Transfer



Demand transfer

- Only requests (level detection) from the external pin (DREQ) are accepted. Activation by other sources is disabled.
- Access to an external area is required (since access to an external area becomes the next activation source).
- The number of blocks is always 1, regardless of the settings.
- Interrupt clear and the DMA interrupt are issued when transfer for the number of times specified is completed.

MB91460 Series

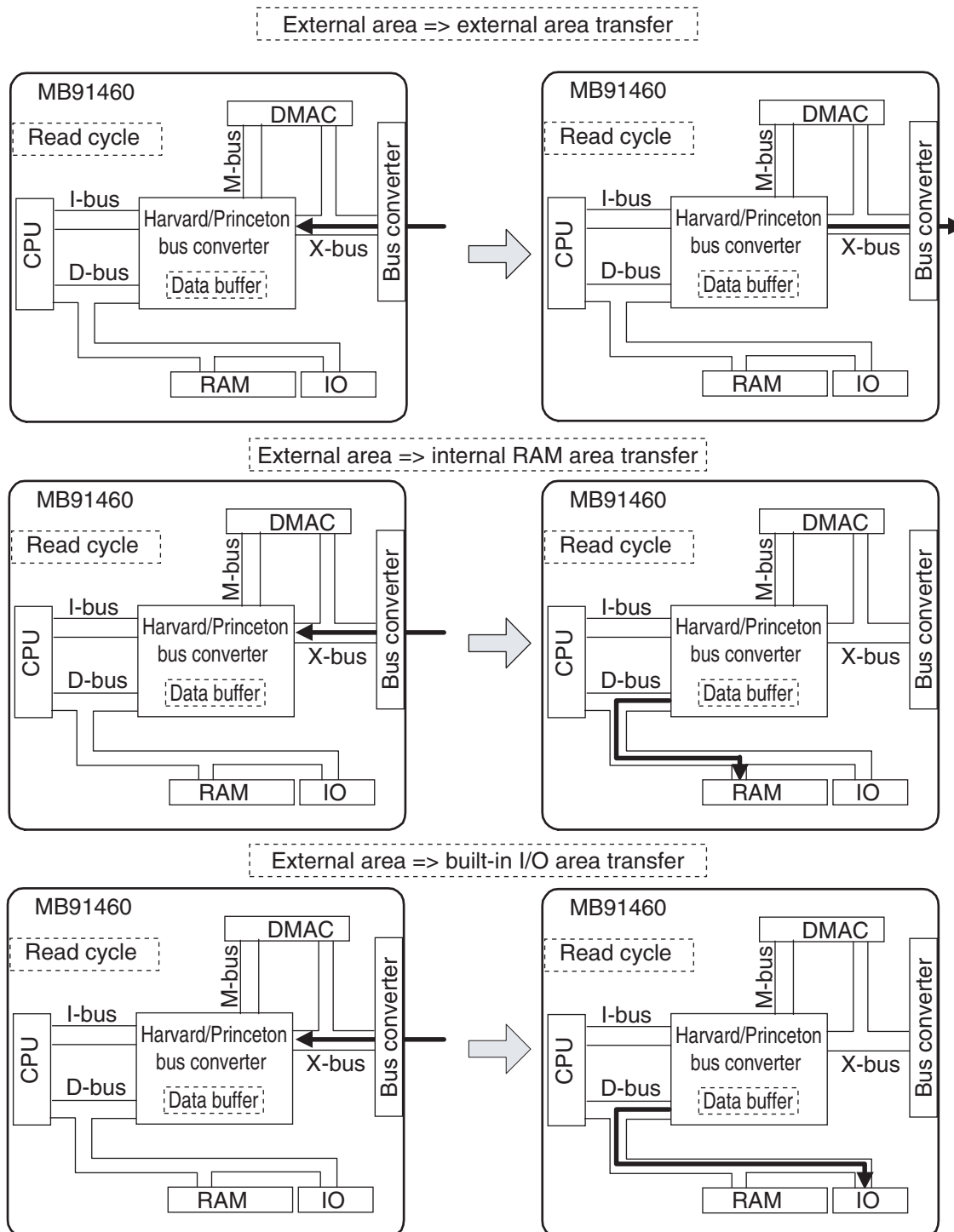
26.5. Data Bus

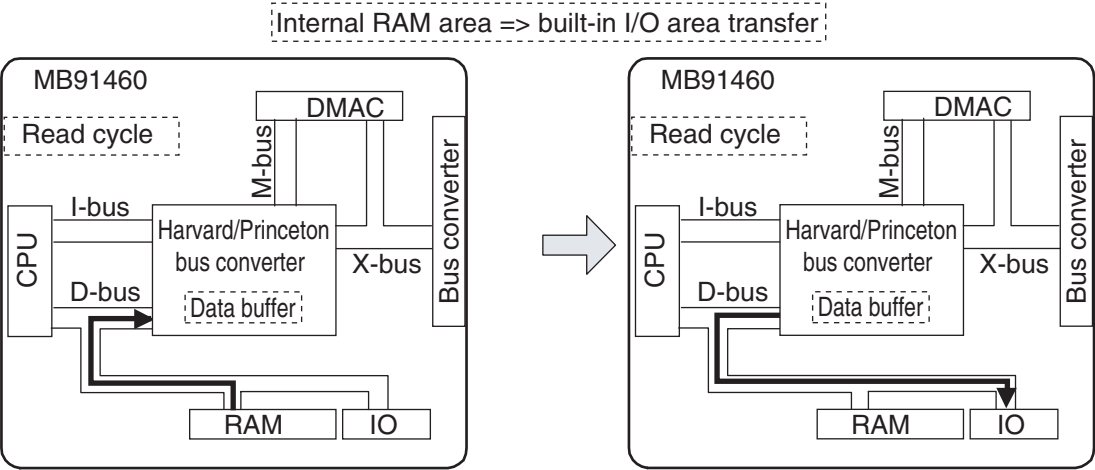
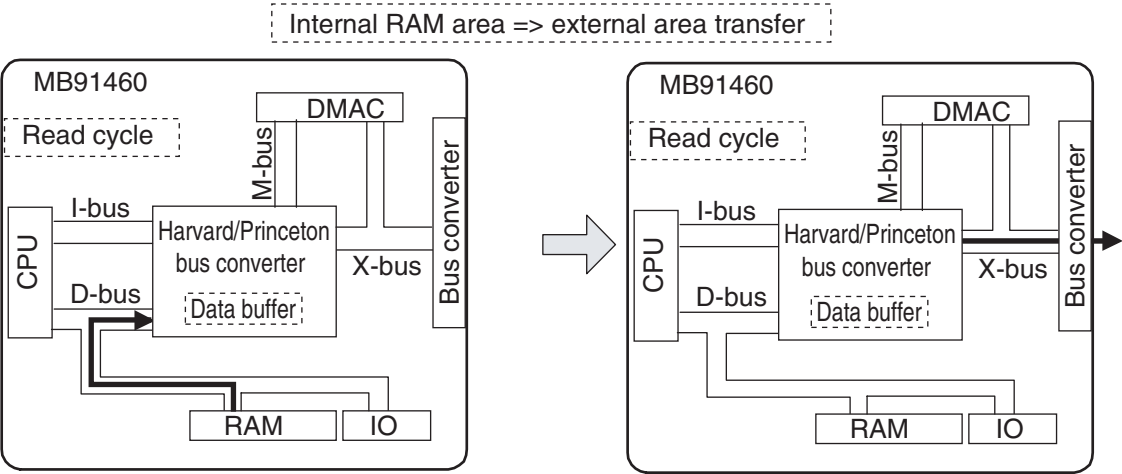
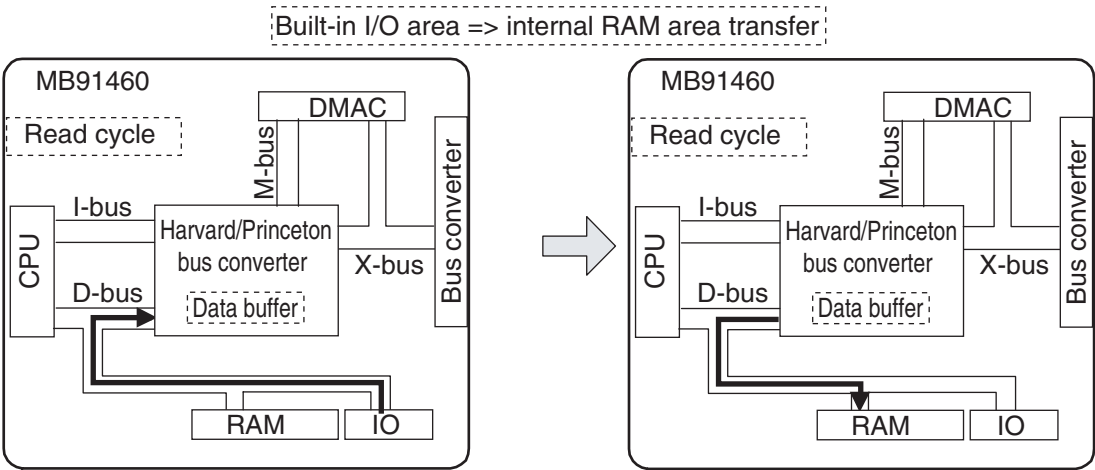
This section shows the flow of data during 2-cycle transfer and fly-by transfer.

■ Flow of Data During 2-Cycle Transfer

Figure 14.5-1 shows examples of six types of transfer during 2-cycle transfer.

Figure 26.5-1 Examples of 2-Cycle Transfer (Continued on next page)



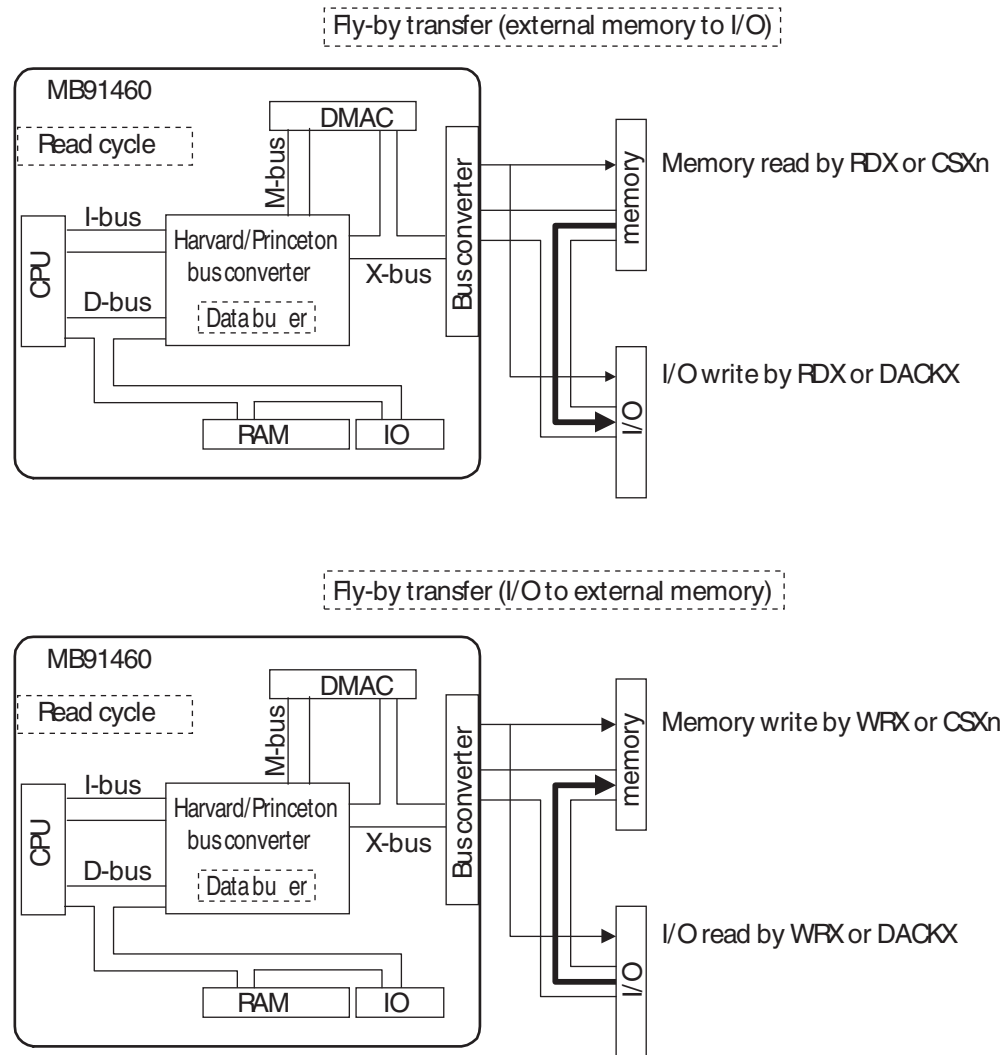


MB91460 Series

■ Flow of Data During Fly-By Transfer

Figure 26.5-2 "Examples of Fly-By Transfer" shows examples of two types of transfer during fly-by transfer.

Figure 26.5-2 Examples of Fly-By Transfer



26.6. DMA External Interface

This section provides operation timing charts for the DMA external interface.

■ DMA External Interface Pins

DMA channels 0-3 have the following DMA-dedicated pins (DREQ, DACKX, DEOTX, and DEOP):

- DREQ: DMA transfer request input pin for demand transfer. A transfer is requested with an input.
- DACKX: This pin becomes active (L output) when DMA accesses an external area via the external interface.
- DEOP: This pin becomes active in synchronization with the last access to complete DMA transfer.
- DEOTX: This pin is used to force the DMA transfer to stop.
- IORDX: This signal becomes active when the direction external I/O -> external memory is selected for fly-by transfer.
- IOWRX: This signal becomes active when the direction external memory -> external I/O is selected for fly-by transfer.

Note:

Refer to 4.10 "DMA Access Operation" for the operation example of DMA external interface.

26.6.1 Input Timing of the DREQ Pin

The DREQ pin is a DMA start request signal. If the pin is also used as a port, enable the DREQ input using the PFR register. This section shows the input timing of the DREQ pin.

■ Timing of Transfer Other Than Demand Transfer

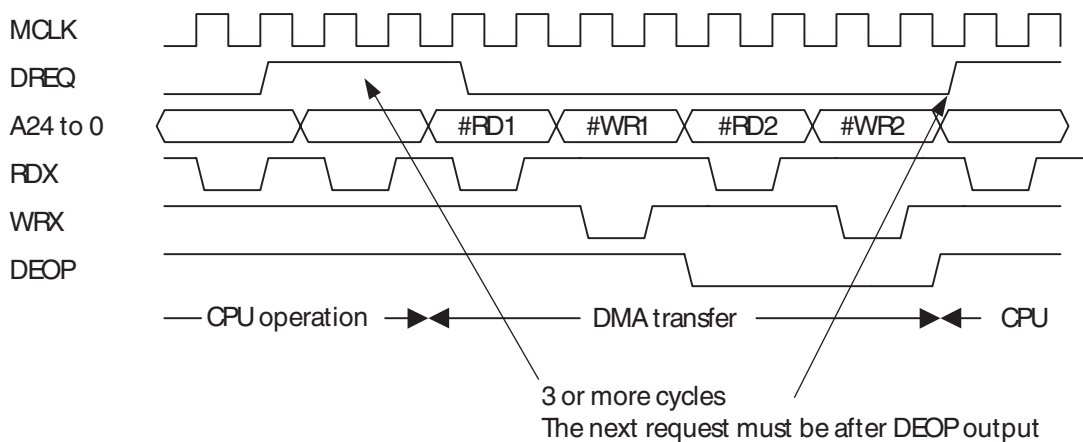
For transfer other than demand transfer, set the DMA start source to edge detection. Although there is no rule for rise/fall timing, use three or more clock cycles as the holding time the DREQ signal. To make another transfer request, enter the request after the DMA transfer is completed (make a request after DEOP is output).

If a request is made before DEOP is output, it may be ignored.

Figure 26.6-1 "Timing Chart for Transfer Other Than the Demand Transfer" shows the timing chart for transfer other than demand transfer.

Figure 26.6-1 Timing Chart for Transfer Other Than the Demand Transfer

When a DREQ edge is requested (for 2-cycle transfer)



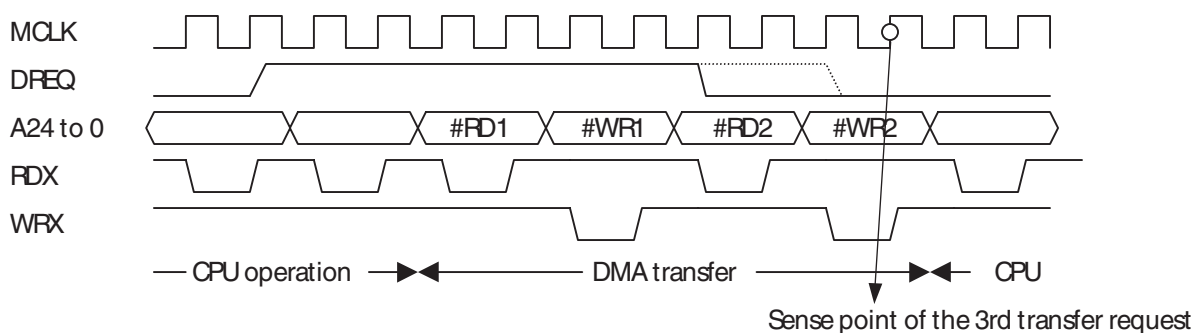
■ Timing of Demand Transfer

For demand transfer, set the DMA start source to level detection. Although there is no rule for starting, synchronize with RDX/WRXn of the DMA transfer when stopping a transfer. The sense timing is the rise of MCLK in the final external access.

Figure 26.6-2 "Timing Chart for Demand Transfer" shows the timing chart for demand transfer.

Figure 26.6-2 Timing Chart for Demand Transfer

When a DREQ level is requested (for 2-cycle transfer)



Note:

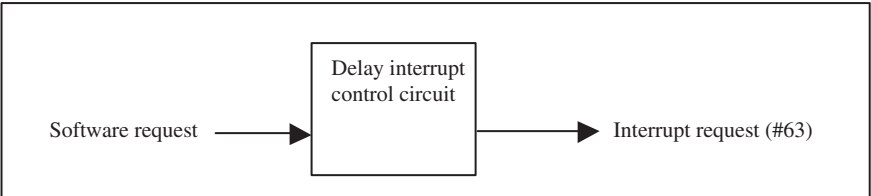
In this case, because 2-cycle transfer is used and the transfer source and transfer destination are an external area, negate from the fall of #RD2 to before the final MCLK rise of #WR2 to stop the two DMA transfer operations.

Chapter 27 Delayed Interrupt

27.1. Overview

The delayed interrupt, or the delayed interrupt module is used to generate an interrupt for task switching.

Figure 27.1-1 Block diagram of delayed Interrupt



27.2. Features

- Type: Interrupt request bit (There is no interrupt request enable bit)
- Quantity: 1
- Other:
 - The software generates/releases interrupt request.
 - Real time OS (REALOS) uses the delayed interrupt for task switching.

27.3. Configuration

Figure 27.3-1 Configuration diagram of delayed interrupt

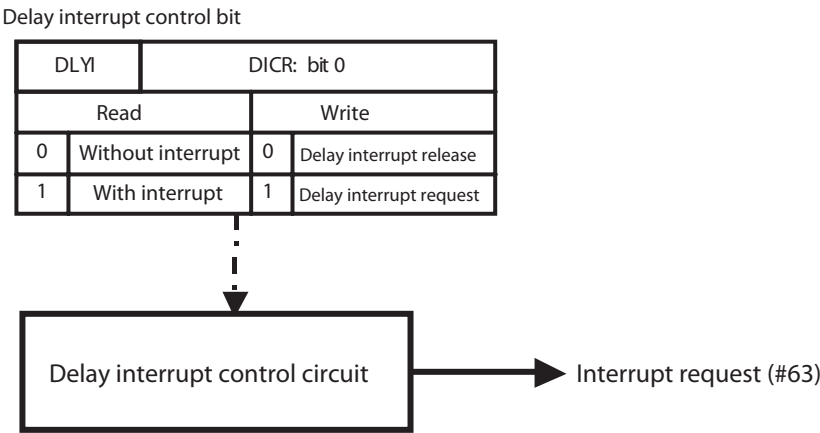


Figure 27.3-2 List of Registers

Delayed interrupt											
Address	Bit	7	6	5	4	3	2	1	0		
00038H		---	---	---	---	---	---	---	DLYI		
										DICR	(Delayed interrupt control)
00457H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0		
										ICR23	(Interrupt level)
0046FH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0		
										ICR47	(Interrupt level)
Address											
0FFF00H										32Bits	(Interrupt vector #63)

27.4. Register

27.4.1 DICR: Delayed Interrupt Control Register

This register controls to generate/clear the delayed interrupt.

- **DICR: Address 0038h (Access: Byte)**

7	6	5	4	3	2	1	0	bit
—	—	—	—	—	—	—	DLYI	
—	—	—	—	—	—	—	0	Initial value
RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	R/W	Attribute

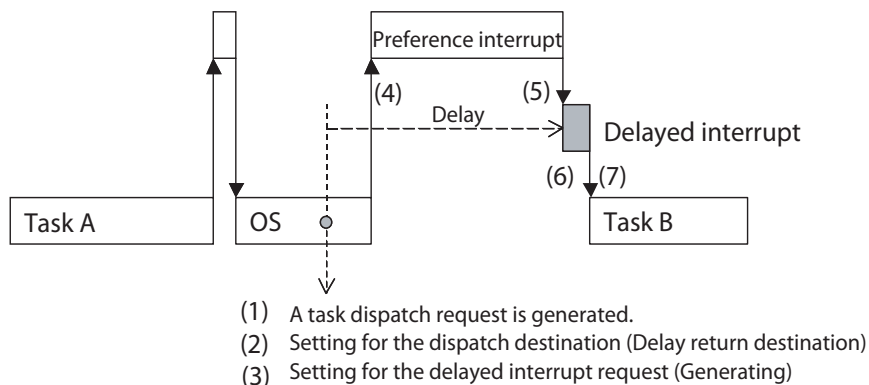
(Refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for the attributes.)

- Bit7-1: Undefined: Writing does not affect operation. The read value is undefined.
- Bit0: Delayed interrupt control bit

DLYI	Read operation	Write operation
0	No delayed interrupt request	Delayed interrupt request clear
1	Delayed interrupt request	Delayed interrupt request generation

27.5. Operation

Figure 27.5-1 Delayed interrupt service



- (1) In OS, a request for task B dispatch is generated
- (2) OS sets the delayed interrupt return destination (dispatch destination)
- (3) OS sets the delayed interrupt (delayed interrupt request generation)
- (4) When OS returns, the interrupt with the highest priority sequence takes place, because an interrupt service is prohibited in OS
- (5) When the interrupt with the highest priority is completed, delayed interrupt takes place
- (6) In delayed interrupt, delayed interrupt is released
- (7) Returned from the delayed interrupt (dispatched to task B)

MB91460 Series

27.6. Setting

Table Setting required for the delayed interrupt generation/clear

Table 27.6-1 Setting required for the delayed interrupt generation/clear

Setting	Setting register	Setting method*
Vector for delayed interrupt	Refer to “ Chapter 24 Interrupt Control (Page No.429) ”	Refer to 24.4.1
Delayed interrupt setting. Generating interrupt request/Releasing interrupt request	Delayed interrupt control register (DICR)	Refer to 27.4.1

*: Refer to the number for the setting method.

27.7. Q & A

27.7.1 What are interrupt-associated registers?

Setting for the delayed interrupt vector and interrupt level

The relationship between the delayed interrupt level and the delayed interrupt vector is shown in the following table.

Refer to “[Chapter 24 Interrupt Control \(Page No.429\)](#)” for more information on the interrupt level and interrupt vector.

Interrupt vector (default)	Interrupt level setting bit (ICR[4:0])
#63 Address: 0FFF00h	Interrupt level register (ICR23) Address: 00457h

The interrupt request bit (DICR.DLYI) cannot automatically be released, and it should be released by the software before returning from an interrupt service. (“0” is written for DLYI bit)

Remark: For REALOS compatibility reasons, ICR23 and ICR47 can be exchanged by setting the REALOS compatibility bit (addr 0x0C03 : IOS[0]) if necessary.

27.7.2 How is the interrupt request generated/cleared?

The delayed interrupt request bit (DICR.DLYI) performs this function.

	Interrupt request enable bit (DLYI)
Clearing an interrupt request	Sets the value to “0”
Generating an interrupt request	Sets the value to “1”

The delayed interrupt does not have an interrupt request enable bit.

27.8. Caution

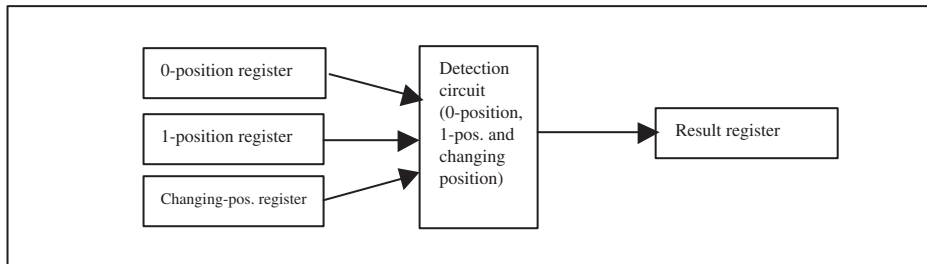
- The delayed interrupt request bit is the same as general interrupt flags. It should be used to clear delayed interrupt request bit in an interrupt routine in addition to switching tasks.
- The delayed interrupt function is used by real time OS (REALOS). As a result, the delayed interrupt function is prohibited in a piece of user software when using real time OS.

Chapter 28 Bit Search

28.1. Overview

The bit search module is used to detect the first '0' position, the first '1' position or the first changing position for data written in specific registers.

Figure 28.1-1 Block diagram of the Bit search module



28.2. Features

- Function: Detects the first position ("1"-position, "0"-position or changing position) by scanning data written in data register from MSB to LSB.
 - 0 detection Detects the first '0' position.
 - 1 detection Detects the first '1' position.
 - Changing position detection Detects the first position where data changes from '0' to '1' or vice versa.
- Quantity: 1

28.3. Configuration

Figure 28.3-1 Configuration Diagram of the bit search module

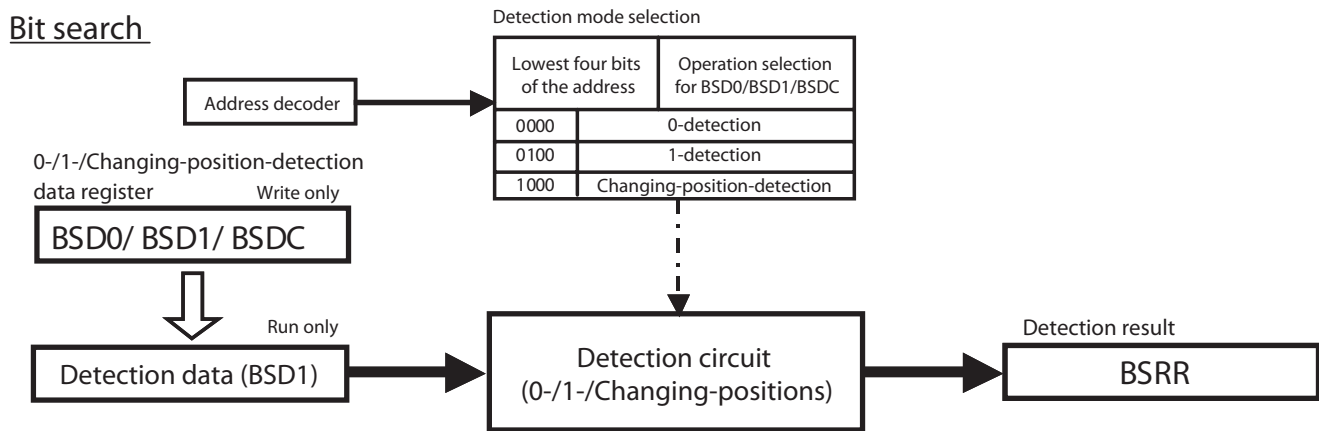


Figure 28.3-2 List of Registers

Bit search

Address			
003F0H	32Bits	BSD0	(0 detection data)
003F4H	32Bits	BSD1	(1 detection data)
003F8H	32Bits	BSDC	(Changing point detection data)
003FCH	32Bits	BSRR	(Detection result)

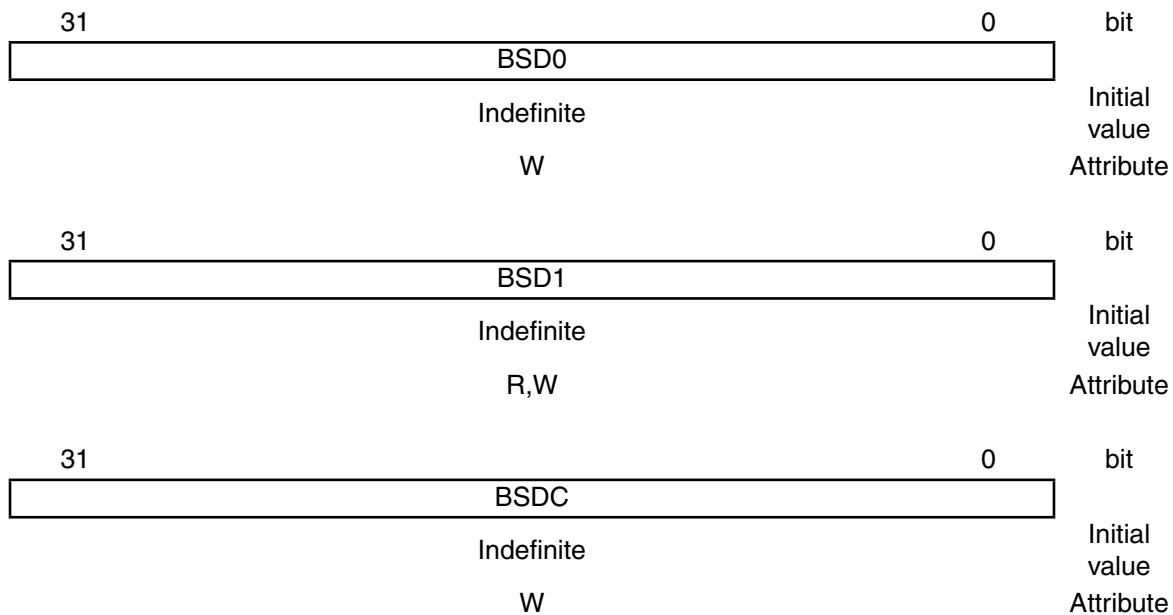
MB91460 Series

28.4. Register

28.4.1 BSD0: 0 Detection Register / BSD1:1 Detection Register / BSDC: Changing position Detection Data Register

This is a register for setting the bit search detection data.

- **BSD0: Address 03F0_H (Access: Word)**
- **BSD1: Address 03F4_H (Access: Word)**
- **BSDC: Address 03F8_H (Access: Word)**



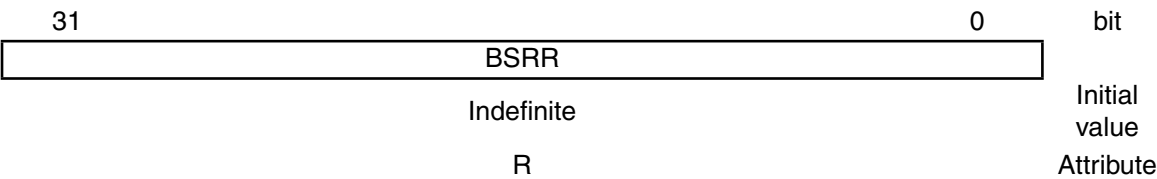
(For the attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- Write data used to detect 0, 1 and changing position in each of the registers BSD0, BSD1 and BSDC.
- The result is stored in the detection result register BSRR.
 - During 0 detection, the position where “0” is first detected is stored for data written in the order of MSB(bit31) to LSB(bit0).
 - During 1 detection, the position where “1” is first detected is stored for data written in the order of MSB(bit31) to LSB(bit0).
 - During change position detection, the position where a value different from MSB(bit31) is first detected is stored for data written in the order of bit30 to LSB(bit0).
- The register BSD0 used for 0 detection and the BSRC register used for changing position detection are write-only. The value during read operation is indefinite.
- Data saved in the bit search can be read if the register BSR1 used to detect 1 is read.
Previous detection result can be restored by re-writing previously read data in the BSR1 used for detecting 1. This applies to the processes for the 0 detection and the changing position detection. This function can be used to restore a specific state when using a bit search in processing such as interrupt handler.

28.4.2 BSRR: Detection Result Register

This register is used to read a bit search result.

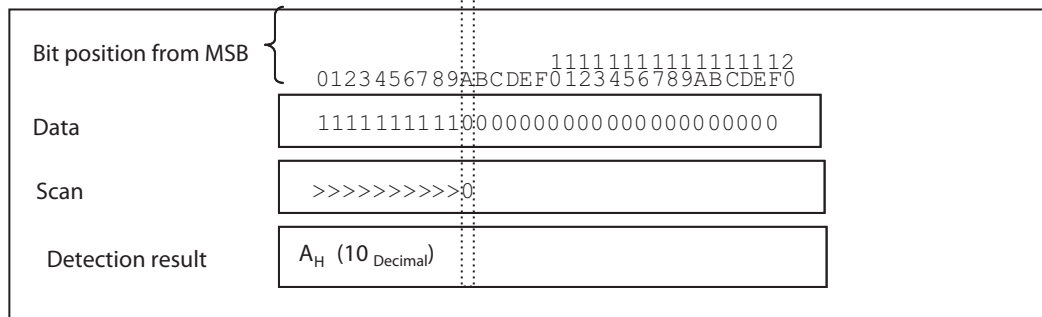
- **BSRR: Address 03FC_H (Access: Word)**



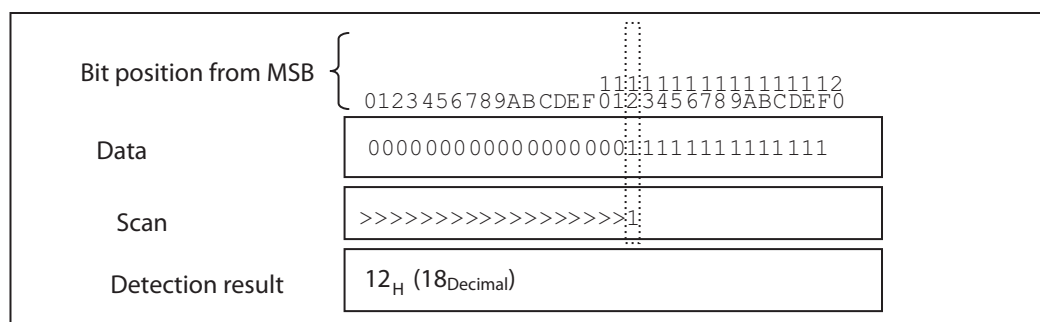
(For the attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- Detection result for data written in the 0 detection register BSD0, the 1-detection register BSD1 and the changing-position-detection register BSDC can be read. Data last written can be read. However, the type of result cannot be identified: Information on 0 detection,1 detection or changing position detection is not included.
A 0 can be read at detection position bit31(MSB), and continues reading 31 at detection position bit0(LSB) by adding 1 at the next position toward bit0(LSB). A value of 32 is read when not detected.
- The detection result register is read-only, and a write operation has no effect.

28.5.1 Zero detection



Write data		Read value (Decimal notation)
1111111111111111000000000000 _B (FFFFF000 _H)	→	20
11111000010010011110000010101010 _B (F849E0AA _H)	→	5
10000000000000001010101010101010 _B (8002AAAA _H)	→	1
11111111111111111111111111111111 _B (FFFFFFFF _H)	→	32

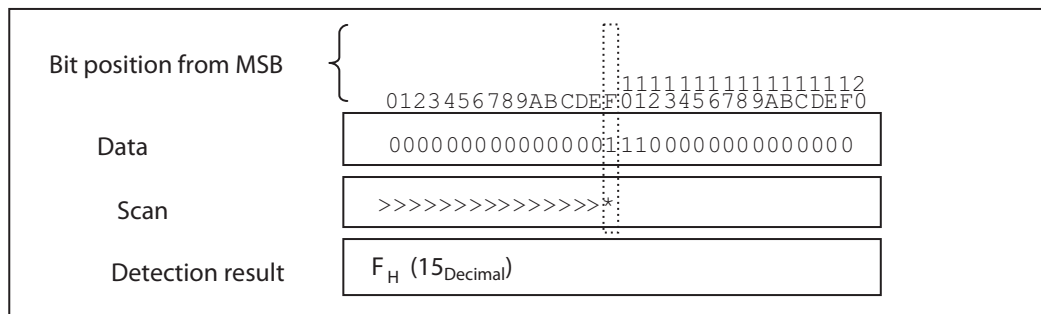


Write data	Read value (Decimal notation)
00100000000000000000000000000000 _B (20000000 _H)	2
00000001001000110100010101100111 _B (01234567 _H)	7

00000000000000001111111111111111 _B	(0003FFFF _H)	→	14
000000000000000000000000000000001 _B	(00000001 _H)	→	31
000000000000000000000000000000000 _B	(00000000 _H)	→	32

28.5.3 Changing Position Detection

Figure 28.5-3 Example of changing position detection



- (1) Bit position from MSB
- (2) Written data (Detection starts once data is written.)
- (3) Detects the changing position by scanning from MSB.
- (4) Detected bit position
- (5) Detection result

A value of '32' is returned as detection result if changing position does not exist.

A value of '0' is not returned as detection result for changing position detection.

- Execution example

Write data		Read value (Decimal notation)
00100000000000000000000000000000 _B (20000000 _H)	→	2
00000001001000110100010101100111 _B (01234567 _H)	→	7
00000000000000011111111111111111 _B (0003FFFF _H)	→	14
00000000000000000000000000000001 _B (00000001 _H)	→	31
00000000000000000000000000000000 _B (00000000 _H)	→	32
11111111111111111111000000000000 _B (FFFFF000 _H)	→	20
11111000010010011110000010101010 _B (F849E0AA _H)	→	5
10000000000000101010101010101010 _B (8002AAAA _H)	→	1
11111111111111111111111111111111 _B (FFFFFFFF _H)	→	32

Table 28.5-1 The Relationship Between the Bit Position and the Value to be Returned (Decimal Notation)

[illegible]

28.6. Setting

Table 28.6-1 Settings Required for “Zero” Position Detection

Setting	Setting register	Setting method *
Data write & scan start	“Zero” position detection data register (BSD0)	Refer to 28.4.
Converted value read	Detection result register (BSRR)	Refer to 28.4.2

*: For detailed description contents, refer to the reference destination number.

Table 28.6-2 Setting Required for Using “One” Position Detection

Setting	Setting register	Setting method *
Data write & scan start	“One” position detection data register (BSD1)	Refer to 28.4.
Converted value read	Detection result register (BSRR)	Refer to 28.4.2

*: For detailed description contents, refer to the reference destination number.

Table 28.6-3 Setting Required for Using Changing Position Detection

Setting	Setting register	Setting method *
Data write & scan start	Changing position detection data register (BSDC)	Refer to 28.4.
Converted value read	Detection result register (BSRR)	Refer to 28.4.2

*: For detailed description contents, refer to the reference destination number.

28.7. Q & A

28.7.1 How is data written?

Writes data with the detection data registers (BSD0, BSD1, BSDC).

Operation mode	Detection data register
"Zero" position detection write	Writes data in (BSD0)
"One" position detection write	Writes data in (BSD1)
Changing position detection write	Writes data in (BSDC)

28.7.2 How is scanning started?

Scanning is started once data is written in the detection data registers (BSD0, BSD1, BSDC).

28.7.3 How is a result read?

The detection result register (BSRR) is read.

28.7.4 How is the previous bit search state restored?

The following restoration processes are performed.

If the previous bit search state should be restored after a bit search has been executed in an interrupt handler.

- 1) Reads data from the one detection data register, and saves the contents. (evacuation)
- 2) The Bit search is used.
- 3) Writes data evacuated in Item 1) in the one detection data register. (restoration)

Using the above procedures, the value to be read next from the detection result register is the one that was written in the bit search executed in 1) or before.

The bit search state can be correctly restored using the above procedures even if the 0-detection, 1-detection or changing-position-detection data register has been written last.

MB91460 Series

28.8. Caution

The following are the remarks on using the bit search module.

- The macros are for REALOS(OS), and the user cannot use them when using REALOS.
- If the relevant detection is not found, a detection result of 32(decimal), 10(hexadecimal) or 10000(binary) is returned.
- A value of “0” is not returned for the changing position detection.
- The data registers (0-detection/1-detection/ changing-position-detection) is a write-only, and accessed by word.
However, the 1-detection read address is assigned to an internal data register for restoration so that restoring previous bit search state is possible. (Refer to “[28.7.3 How is a result read? \(Page No.528\)](#)”.)
- The 0-detection register BSD0, 1-detection register BSD1 and changing-position-detection register BSDC are included in one register in terms of the structure. The operation is selected with the lowest four bits of the accessing address.

MB91460 Series

Chapter 29 MPU / EDSU

29.1. Overview

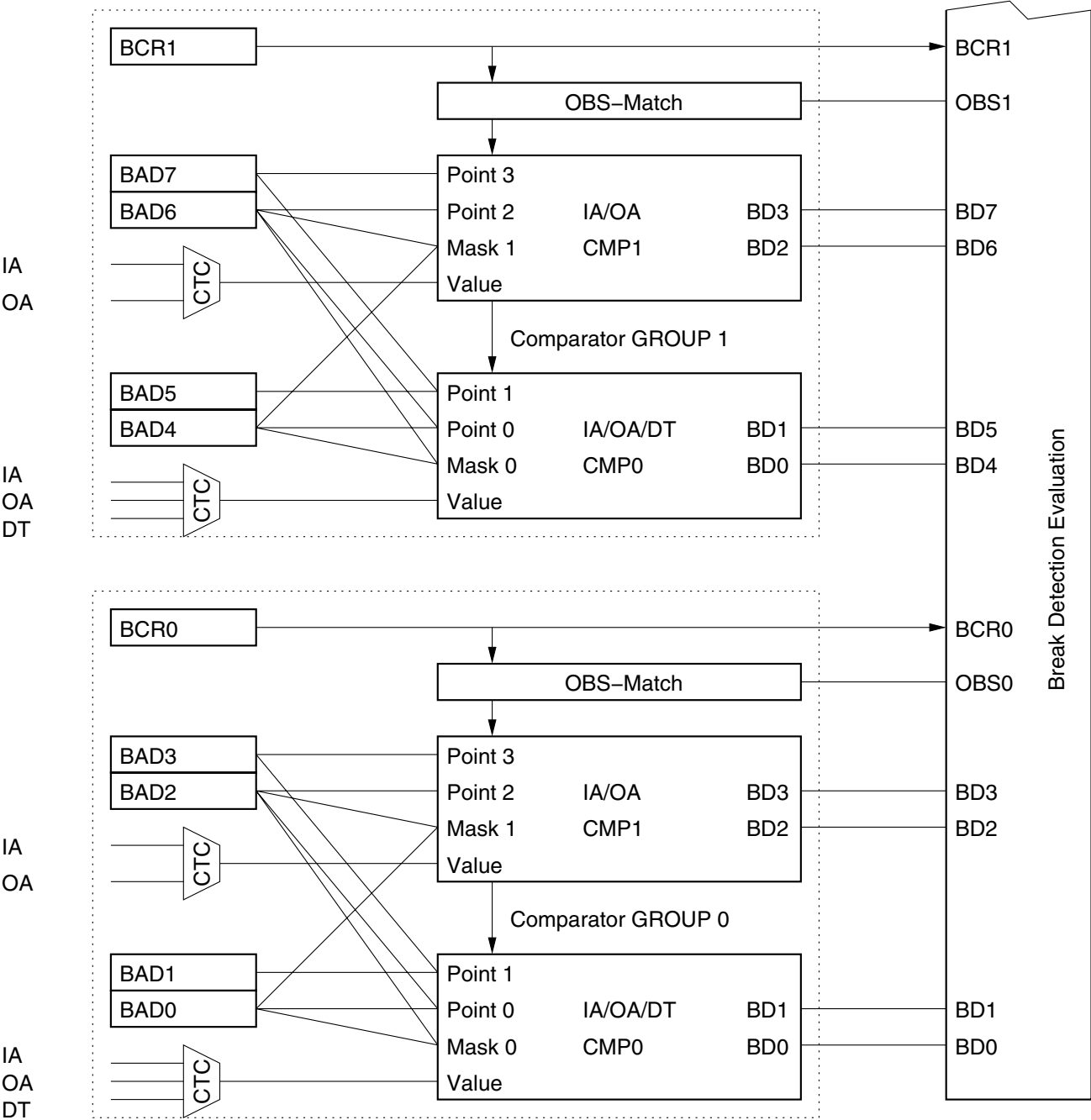
Memory Protection Unit (MPU) and Embedded Debug Support Unit (EDSU) for MB91460 series.

Note: The MPU/EDSU module features a clock disable function. For enabling the MPU/EDSU module it is necessary to set the EDSUEN bit in the CSCFG register. See chapter “[CSCFG: Clock Source Configuration Register \(Page No.299\)](#)” for further information.

The features are scalable in units of 'Comparator Groups'. The number of this Groups can be defined from one to eight. Features of one Comparator Group are listed below:

- A total number of 4 Breakpoints, could be programmed to:
 - 4 Instruction Address Breakpoints
 - 4 Operand Address Breakpoints (programmable on datasize and access type)
 - 2 Operand Address Breakpoints and 2 Instruction Address Breakpoints
 - 2 Operand Address Breakpoints and 2 Data Value Breakpoints
- 2 Masks possible to assign (reduces the number of breakpoints)
- 2 Range Functions
- Break Trigger programmable on resource interrupts
- MPU functionality
 - User and Supervisor permission for read/write/execute
 - Default permissions for the whole MCU address range
 - Permission definition for two address ranges per Comparator Group (8 Groups result in 16 MPU Channels)
 - Can detect DMA accesses on the D-Bus and Resource address regions
 - Register set is locked in User mode
 - Dynamic configuration possible, privileged configuration with INT #5 is not interruptible
 - A permission violation causes an MPUPV trap
- Capture register for Instruction Address and Operand Address (for MPU and Operand Break)
- Capture information for MPU channel index, DMA flag, Operand Size and Access Type

Figure 29.1-1 EDSU block diagram



MB91460 Series

29.2. Features

One Comparator Group offers up to 4 Breakpoints. One Group consists of two full-featured range comparators with the option to use two point registers as mask information. The following features could be partially mixed-up:

- 4 Instruction Address Breakpoints

Up to 4 instruction address breakpoints can be defined.

Two instruction breakpoints can be masked. The other two registers can operate as mask registers then. Also maskable is a break address range made with two points and one mask register.

Two absolute address ranges for instruction breakpoints can be defined where 2 or 4 out of 4 instruction breakpoint registers are assigned for the range.

- 4 Operand Address Breakpoints

Up to 4 operand address breakpoints can be defined.

Two operand breakpoints can be masked. The other two registers can operate as mask registers then. Also maskable is a break address range made with two points and one mask register.

Two absolute address ranges for operand breakpoints can be defined where 2 or 4 out of 4 operand breakpoint registers are assigned for the range.

Operand breaks can be selected for datasizes: byte, halfword and word on access types: read, read-modify-write and write.

- 2 Operand Data Value Breakpoints

Up to 2 operand data value breakpoints can be defined.

The definition of one data value range is possible.

One data value breakpoint can be masked by defining the other point as mask register.

The Operand Address and Data Value Breakpoints can be switched to a combined trigger condition.

- Memory protection

Two channels/ranges could be defined to operate in memory protection mode.

Possible is the protection of two Operand Address ranges, two Instruction Address Ranges or a combination of one Operand and one Instruction Address range.

Read/write or execute permission could be defined for each channel, both for the normal User and the Supervisor mode.

29.3. Registers

29.3.1 List of EDSU Registers

Table 29.3-1 EDSU Registers Summary

Address	Register				Block
	+0	+1	+2	+3	
F000 _H	BCTRL [R/W] ----- ----- 11111100 00000000				EDSU
F004 _H	BSTAT [R/W] ¹ ----- -----000 00000000 10--0000				
F008 _H	BIAC [R] 00000000 00000000 00000000 00000000				
F00C _H	BOAC [R] 00000000 00000000 00000000 00000000				
F010 _H	BIRQ [R/W] 00000000 00000000 00000000 00000000				
F014 _H ...F01F _H	reserved -				
F020 _H	BCR0 [R/W] ----- 00000000 00000000 00000000				
F024 _H	BCR1 [R/W] ----- 00000000 00000000 00000000				
F028 _H	BCR2 [R/W] ----- 00000000 00000000 00000000				
F02C _H	BCR3 [R/W] ----- 00000000 00000000 00000000				
F030 _H	BCR4 [R/W] ----- 00000000 00000000 00000000				
F034 _H	BCR5 [R/W] ----- 00000000 00000000 00000000				
F038 _H	BCR6 [R/W] ----- 00000000 00000000 00000000				
F03C _H	BCR7 [R/W] ----- 00000000 00000000 00000000				
F040 _H ...F07F _H	reserved -				

Table 29.3-1 EDSU Registers Summary

Address	Register				Block
	+0	+1	+2	+3	
F080 _H	XXXXXXXX	BAD0 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	EDSU
F084 _H	XXXXXXXX	BAD1 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F088 _H	XXXXXXXX	BAD2 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F08C _H	XXXXXXXX	BAD3 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F090 _H	XXXXXXXX	BAD4 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F094 _H	XXXXXXXX	BAD5 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F098 _H	XXXXXXXX	BAD6 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F09C _H	XXXXXXXX	BAD7 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0A0 _H	XXXXXXXX	BAD8 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0A4 _H	XXXXXXXX	BAD9 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0A8 _H	XXXXXXXX	BAD10 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0AC _H	XXXXXXXX	BAD11 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0B0 _H	XXXXXXXX	BAD12 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0B4 _H	XXXXXXXX	BAD13 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0B8 _H	XXXXXXXX	BAD14 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0BC _H	XXXXXXXX	BAD15 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	

Table 29.3-1 EDSU Registers Summary

Address	Register				Block
	+0	+1	+2	+3	
F0C0 _H	XXXXXXXX	BAD16 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	EDSU
F0C4 _H	XXXXXXXX	BAD17 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0C8 _H	XXXXXXXX	BAD18 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0CC _H	XXXXXXXX	BAD19 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0D0 _H	XXXXXXXX	BAD20 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0D4 _H	XXXXXXXX	BAD21 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0D8 _H	XXXXXXXX	BAD22 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0DC _H	XXXXXXXX	BAD23 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0E0 _H	XXXXXXXX	BAD24 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0E4 _H	XXXXXXXX	BAD25 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0E8 _H	XXXXXXXX	BAD26 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0EC _H	XXXXXXXX	BAD27 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0F0 _H	XXXXXXXX	BAD28 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0F4 _H	XXXXXXXX	BAD29 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0F8 _H	XXXXXXXX	BAD30 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	
F0FC _H	XXXXXXXX	BAD31 XXXXXXXX	[R/W] XXXXXXXX	XXXXXXXX	

1. RMW - read returns '1' for each flag, for write only '0' (clear) is supported.

Note: Read and write access to all registers is byte, halfword and word.

MB91460 Series

29.3.2 Explanations of Registers

■ BCTRL (EDSU Control Register)

BCTRL byte 2	15	14	13	12	11	10	9	8	← Bit no.
Address : F002H	SR	SW	SX	UR	UW	UX	FCPU	FDMA	
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	

BCTRL byte 3	7	6	5	4	3	2	1	0	← Bit no.
Address : F003H	EEMM	PFD	SINT1	SINT0	EINT1	EINT0	EINTT	EINTR	
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

● Default Permission Register

The default permission register defines the lowest priority access permission for the whole memory and I/O address range of the MCU. Lowest priority means, that the default permission take effect for all address regions, which are NOT covered by any dedicated channel configuration, operating in MPU mode. Default read, write and execute permission could be defined for the supervisor mode (SV=1) and the normal user mode (SV=0). The supervisor mode (SV) is indicated by bit 6 of the CCR in the program status word of the CPU. After the INIT condition all permissions are set (access allowed).

BIT[15]: SR - Supervisor default Read permission bit

0	Supervisor is not permitted to read data
1	Supervisor is permitted to read data (default)

BIT[14]: SW - Supervisor default Write permission bit

0	Supervisor is not permitted to write data
1	Supervisor is permitted to write data (default)

BIT[13]: SX - Supervisor default eXecute permission bit

0	Supervisor is not permitted to execute code
1	Supervisor is permitted to execute code (default)

BIT[12]: UR - User default Read permission bit

0	User is not permitted to read data
1	User is permitted to read data (default)

BIT[11]: UW - User default Write permission bit

0	User is not permitted to write data
1	User is permitted to write data (default)

BIT[10]: UX - User default eXecute permission bit

0	User is not permitted to execute code
1	User is permitted to execute code (default)

● CPU and DMA Filter Option Register

BIT[9]: FCPU - Filter CPU access

0	Trigger on CPU accesses (default)
1	Do not trigger on CPU accesses

FCPU controls the filter operation for CPU accesses triggered by operand compare channels (Operand address break, data value break and memory data protection).

If FCPU is set to '1', all CPU accesses are masked out. If set to '0' CPU accesses can cause break function.

BIT[8]: FDMA - Filter DMA access

0	Trigger on DMA accesses (default)
1	Do not trigger on DMA accesses

FDMA controls the filter operation for DMA accesses triggered by operand compare channels (Operand address break, data value break and memory data protection).

If FDMA is set to '1', all DMA accesses are masked out. If set to '0' DMA accesses can cause break function.

Important Note for FDMA: Only DMA accesses over D-Bus were detected. The operands for an explicit DMA trigger condition have to be located in the D-Bus address area (This is the case for D-bus RAM, CAN and all R-Bus resources in the MB91460 family). Otherwise the DMA transfer could not be recognized by the EDSU. This function was mainly intended to disable the trigger on DMA accesses (filter out the operand change condition by DMA), complete address range DMA trigger conditions are not supported.

MB91460 Series

● Enable Interrupt Register

BIT[7]: EEMM - Enable Emulation Mode

0	Disable emulation mode (default)
1	Enable emulation mode

If EEMM is set to '1' then the emulation mode is entered during Step Trace Mode and EDSU exceptions Instruction Break, Operand Break and Tool NMI. During emulation mode the Watchdog Timer (WDT) is disabled. EDSU triggered emulation mode is left with the RETI instruction.

Set to '0' disables emulation mode function. The WDT is not stopped during Step Trace and EDSU exceptions.

BIT[6]: PFD - Phantom Filter Disable

0	Instruction break detection uses phantom filter (default)
1	Phantom Filter disabled

The default (PFD=0) is to use policies to filter out phantom interrupts and wrong status bits, which may be set in addition.

- The instruction fetched, after RETI was executed, is normally the instruction on which the break point was set. Fetch is repeated after processing the breakpoint handler ISR before executing the instruction at the break point. The filter avoids that the trigger of the break condition will be repeated.
- Not granted Instruction Break exceptions are timed out
 - Pre-fetched, but not executed commands
 - Commands after delayed slot instruction
- Consecutive break conditions which are pre-fetched are not allowed to set flags. Only the instruction at which the break condition occurs at first time can set status bits accordingly.
- Nested Instruction breaks are not allowed (break within the break handler ISR)

BIT[5:4]: SINT[1:0] - Select resource INTerrupt source

SINT1 and SINT0 select the active resource interrupt source.

SINT[1:0]	MB91V460A	Resource
00	Tool NMI by interrupt on source 0 selected (default)	UART 0 RX / UART 0 TX
01	Tool NMI by interrupt on source 1 selected	UART 1 RX / UART 1 TX
10	Tool NMI by interrupt on source 2 selected	UART 2 RX / UART 2 TX
11	Tool NMI by interrupt on source 3 selected	CAN 0 / CAN 1

SINT[1:0]	MB91FV460B and all derivatives	Resource
00	Tool NMI by interrupt on source 0 selected (default)	UART 2 RX / UART 2 TX
01	Tool NMI by interrupt on source 1 selected	UART 4 RX / UART 4 TX
10	Tool NMI by interrupt on source 2 selected	UART 5 RX / UART 5 TX
11	Tool NMI by interrupt on source 3 selected	CAN 0 / CAN 1

BIT[3]: EINT1 - Enable extended INTerrupt 1

0	Disable extended interrupt source 1 (default)
1	Enable extended interrupt source 1

If EINT1 is set to '1' then a Tool NMI will be generated on an extended interrupt event at source channel 1. Set to '0' disables this function.

Note: EINT1 interrupt source is not available for the MB91460 series.

BIT[2]: EINT0 - Enable extended INTerrupt 0

0	Disable extended interrupt source 0 (default)
1	Enable extended interrupt source 0

If EINT0 is set to '1' then a Tool NMI will be generated on an extended interrupt event at source channel 0. Set to '0' disables this function.

Note: EINT1 and EINT0 can be used for indicating a signal line event which can be used for generating a BREAK function. The sources of these interrupts are hardwired in the MCU and can be for example: external interrupt ports, general purpose I/O port pins, other resources, etc. This has to be defined in the device specification.

BIT[1]: EINTT - Enable INTerrupt on Transmit

0	Disable transmit interrupt source channels 0 to 3 (default)
1	Enable transmit interrupt source channels 0 to 3

If EINTT is set to '1' then a Tool NMI will be generated on a transmit interrupt event at source channels 0 to 3 set by TXINT[1:0]. Setting EINTT to '0' disables this function.

Note: If SINT[1:0] is set to "11" this bit enables the interrupt of CAN channel 1 (CAN has one interrupt request for both reception and transmission).

MB91460 Series**BIT[0]: EINTR - Enable INTerrupt on Receive**

0	Disable receive interrupt source channels 0 to 3 (default)
1	Enable receive interrupt source channels 0 to 3

If EINTR is set to '1' then a Tool NMI will be generated on a receive interrupt event at source channels 0 to 3 set by RXINT[1:0]. Setting EINTR to '0' disables this function.

Note: If SINT[1:0] is set to "11" this bit enables the interrupt of CAN channel 0 (CAN has one interrupt request for both reception and transmission).

■ BSTAT (EDSU Status Register)

BSTAT byte 2	15	14	13	12	11	10	9	8	← Bit no.
Address : F006H	IDX4	IDX3	IDX2	IDX1	IDX0	CDMA	CSZ1	CSZ0	
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BSTAT byte 3	7	6	5	4	3	2	1	0	← Bit no.
Address : F007H	CRW1	CRW0	PV	RST	INT1	INT0	INTT	INTR	
Read/write ⇒	(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	

BIT[15:11]: IDX[4:0] - Channel Index Indication of MPUPV Trigger

In the case of triggering a memory protection violation (MPUPV), the index of the channel pair 0...15 is saved in The IDX register, which caused the trigger. The channel pairs are normally used as range comparators.

If no MPU channel has detected a hit on its address range, the default permissions apply. If the default permissions are violated, IDX is set to the value 16 (overrun). If the permissions of a matching MPU channel are violated, IDX shows the index of the appropriate break detection bits BIRQ_BD[31:0]. The break detection bits belonging to this comparator are BD[2*IDX] and BD[2*IDX+1].

In case of multiple range hits and/or trigger conditions, the channel with the highest priority trigger condition is indicated by IDX[4:0]. The priority raises with the channel index.

IDX	Description
0-15	Points to the channel number of the last protection violation
16	The last protection violation was caused by the violation of the default permissions

The channel index indication register can be read only.

● Access Type Capture Register

In case of a trap caused by a memory protection violation or an operand/data value break condition, the status bits [12:8] capture type information about the break causing operand access. In case of a memory protection fault due to the violation of execution permissions, this information is also captured, regardless if there was an active operand access or not.

Access type capture register are read only.

BIT[10]: CDMA - Capture DMA Indication

0	The operand access was executed by the CPU
1	The operand access was executed by the DMA controller

BIT[9:8]: CSZ[1:0] - Capture Operand Size

00	The operand has a bit size of 8
01	The operand has a bit size of 16
10	The operand has a bit size of 32
11	reserved

BIT[7:6]: CRW[1:0] - Capture Operand Access Type

00	The operand has been read
01	The operand has been read by read-modify-write indicated
10	The operand has been written
11	no operand access

BIT[5]: PV - Protection Violation Detection

0	There was no protection violation on read, write and execute permissions
1	A protection violation (MPUPV) has been occurred

If this bit is set after a protection violation, a MPUPV trap is indicated to the CPU. The occurrence of a protection violation means, that there was a read or write access to a defined address region, which was not permitted or code was executed without execute permissions for this address region. As consequence the CPU switches to supervisor mode (SV=1) and calls the handler routine for interrupt number #6 (see [Table 29.4-8 on page 562](#)).

This bit should be cleared by writing '0' in the MPUPV trap handler routine.

BIT[4]: RST - Operation Initialization Reset (RST) Detection

The reset operation of FRex family is divided into two levels, setting initialization reset (INIT) and operation initialization reset (RST). When INIT occurs, RST occurs at the same time implicitly.

0	Operation Reset was not triggered since last BSTAT read or clear
1	Operation Reset was triggered since last BSTAT read or clear

The RST bit is read only, any write access to this bit will be ignored. RST is cleared after BSTAT is read (read from any byte address within the 32 bit word). RST has same behaviour for read and read-modify-write

MB91460 Series

access.

The RST bit can be used for reset detection. It is set in any case of operation initialization reset is triggered. Debug monitor software can use this to detect if the communication device to the debugger front end needs to be re-configured after an operation reset. This is important for debugging of boot procedures and soft reset handling. After reading the EDSU status word the RST bit is cleared automatically.

● Break Interrupt Register

BIT[3]: INT1 - INTerrupt on extended source 1

0	Interrupt on extended source channel 1 not detected (default)
1	Interrupt on extended source channel 1 detected

INT1 reflects the status of the extended interrupt source channel 1. It is set to '1' if a high level on the extended interrupt signal line has been occurred. The status of '1' is stored until cleared by software.

Writing '0' resets the INT1 bit to '0'. Writing '1' to this bit is ignored. On a Read Modify Write instruction INT1 is read as '1'.

BIT[2]: INT0 - INTerrupt on extended source 0

0	Interrupt on extended source channel 0 not detected (default)
1	Interrupt on extended source channel 0 detected

INT0 reflects the status of the extended interrupt source channel 0. It is set to '1' if a high level on the extended interrupt signal line has been occurred. The status of '1' is stored until cleared by software.

Writing '0' resets the INT0 bit to '0'. Writing '1' to this bit is ignored. On a Read Modify Write instruction INT0 is read as '1'.

BIT[1]: INTT - INTerrupt on Transmit source

0	Interrupt on transmit source not detected (default)
1	Interrupt on transmit source channel detected

INTT reflects the status of the transmit interrupt source channels 0 to 3 (can be selected by TXINT[1:0]). It is set to '1' on a high level on the transmit interrupt signal line and '0' on a low level on the signal line.

This bit is read-only. It can be set to '0' by clearing the appropriate interrupt bit in the selected resource.

Note: If SINT[1:0] is set to "11" this bit indicates the interrupt of CAN channel 1 (CAN has one interrupt request for both reception and transmission).

BIT[0]: INTR - INTerrupt on Receive source

0	Interrupt on receive source not detected (default)
1	Interrupt on receive source channel detected

INTR reflects the status of the receive interrupt source channels 0 to 3 (can be selected by RXINT[1:0]). It is set to '1' on a high level on the receive interrupt signal line and '0' on a low level on the signal line.

This bit is read-only. It can be set to '0' by clearing the appropriate interrupt flag in the selected resource.

Note: If SINT[1:0] is set to "11" this bit indicates the interrupt of CAN channel 1 (CAN has one interrupt request for both reception and transmission).

■ BIAC (EDSU Instruction Address Capture Register)

BIAC [R]				
Address	+0	+1	+2	+3
F008 _H	00000000	00000000	00000000	00000000

This register captures the address of the instruction (IA), which has caused the protection violation or the operand/data value break. This register is read only.

Note: In case of Instruction Address (IA), Operand Address (OA) or Operand Data Value Break (DT) the BIAC register keeps valid Instruction Address (IA) until respective BIRQ.BD-bits are reset.

In case of Protection Violation Break (PV) the BIAC register keeps valid Instruction Address (IA) until PV-bit is reset.

In case of multiple breaks PV along with e.g. IA, OA or DT the BIAC register keeps valid Instruction Address (IA) until both PV-bit and respective BIRQ.BD-bits are reset.

■ BOAC (EDSU Operand Address Capture Register)

BOAC [R]				
Address	+0	+1	+2	+3
F00C _H	00000000	00000000	00000000	00000000

This register captures the address of the operand access (OA), which has caused the protection violation or the operand/data value break. This register is read only.

■ BIRQ (EDSU Break Detection Interrupt Request Register)

BIRQ [R/W]				
Address	+0	+1	+2	+3
F010 _H	00000000	00000000	00000000	00000000

BIRQ collects all break detection bits of all channels, regardless of the type configuration of each channel. The actual implementation consists of 8 groups of channels, that are 32 single point channels totally.

Each group of channels consists of 4 channels and 4 bits for break detection in the BIRQ register. Each group has two comparator pairs. Each pair consists of two point comparators which could build a range comparator by setting the range enable bit. Such a range comparator pair is connected to the instruction address, operand address or the data value information - selected by the comparator type configuration.

For detection of combined operand address and data value breaks two of such comparator pairs are combined together. Then the break detection (BD) bits are set only if both conditions are matching simultaneously.

MB91460 Series**BIT[31:0]: BD[31:0] - Break Detection register**

0	Break factor not detected (default)
1	Break factor detected on channel according the bit position [31:0]

BD[31:0] reflects the status of the break detection. It is set to '1' at match with BAD31...BAD0 accordingly (and if the mask condition is satisfied, if enabled by EM1/0). For bit pairs [31:30], [29:28], ..., [1:0] range matches could apply, if the range function using two points is enabled by ER1/0.

Break factors could be

- instruction address break,
- operand address break,
- data value break,
- combined operand address and data value break and
- memory protection violation.

Writing '0' resets the BD[31:0] bits to '0'. Writing '1' to these bits is ignored. On a Read Modify Write instruction all BD bits are read as '1'.

BD1/BD0 setting at enabled address range function (also valid for the other pairs of BD bits in neighbourhood):

If the operand address range function is enabled with ER0 in addition to the point enables EP1 and EP0, then the BD1 and BD0 detection bits are set in the following manner:

Table 29.3-2 BD Coding for Match on Start/Endpoint or Range

BD1	BD0	Compare value: Instruction, Operand Address, Data Value
0	0	No match (Default)
0	1	Match on point (compare value == BAD0)
1	0	Match on point (compare value == BAD1)
1	1	Match on range (BAD0 < compare value < BAD1)

■ BCR0...BCR7 (EDSU Channel Configuration Register0...7)

BCR 0, byte 0	31	30	29	28	27	26	25	24	← Bit no.
Address : F020H	-	-	-	-	-	-	-	-	
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	
Default value⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

BCR 0, byte 1	23	22	21	20	19	18	17	16	← Bit no.
Address : F021H	SRX1	SW1	SRX0	SW0	URX1	UW1	URX0	UW0	
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BCR 0, byte 2	15	14	13	12	11	10	9	8	← Bit no.
Address : F022H	MPE	COMB	CTC1	CTC0	OBS1	OBS0	OBT1	OBT0	
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BCR 0, byte 3	7	6	5	4	3	2	1	0	← Bit no.
Address : F023H	EP3	EP2	EP1	EP0	EM1	EM0	ER1	ER0	
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

For each group of four channels one channel configuration register (BCR0...BCR7) is implemented. It holds the configuration set for the according group of channels. The following table shows the relationship, which channel configuration, break point address/data registers and break detection bits belong together.

Table 29.3-3 Relationship of BCR, BAD and BIRQ registers

Group Config	Address/Data	BADx Usage	Point	Mask	Combination		BIRQ
BCR0	BAD0	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD0
	BAD1	Point1	EP1			IA1 / OA1 / DT1	BD1
	BAD2	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD2
	BAD3	Point3	EP3			IA1 / OA1	BD3
BCR1	BAD4	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD4
	BAD5	Point1	EP1			IA1 / OA1 / DT1	BD5
	BAD6	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD6
	BAD7	Point3	EP3			IA1 / OA1	BD7
BCR2	BAD8	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD8
	BAD9	Point1	EP1			IA1 / OA1 / DT1	BD9
	BAD10	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD10
	BAD11	Point3	EP3			IA1 / OA1	BD11

MB91460 Series**Table 29.3-3 Relationship of BCR, BAD and BIRQ registers**

Group Config	Address/Data	BADx Usage	Point	Mask	Combination		BIRQ
BCR3	BAD12	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD12
	BAD13	Point1	EP1			IA1 / OA1 / DT1	BD13
	BAD14	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD14
	BAD15	Point3	EP3			IA1 / OA1	BD15
BCR4	BAD16	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD16
	BAD17	Point1	EP1			IA1 / OA1 / DT1	BD17
	BAD18	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD18
	BAD19	Point3	EP3			IA1 / OA1	BD19
BCR5	BAD20	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD20
	BAD21	Point1	EP1			IA1 / OA1 / DT1	BD21
	BAD22	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD22
	BAD23	Point3	EP3			IA1 / OA1	BD23
BCR6	BAD24	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD24
	BAD25	Point1	EP1			IA1 / OA1 / DT1	BD25
	BAD26	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD26
	BAD27	Point3	EP3			IA1 / OA1	BD27
BCR7	BAD28	Point0, Mask0	EP0	EM0	range 0 ER0	IA0 / OA0 / DT0	BD28
	BAD29	Point1	EP1			IA1 / OA1 / DT1	BD29
	BAD30	Point2, Mask1	EP2	EM1	range 1 ER1	IA0 / OA0	BD30
	BAD31	Point3	EP3			IA1 / OA1	BD31

● Group of Channels, Permission Definition Register

The permission definition registers are valid only for the group of channels operating in MPU mode. This is the case if MPE is set to '1'. If the group does not operate in MPU mode, the permission configuration is not required (don't care).

Normally MPU channels operate in range mode for the address definitions.

The type of the permission, which could be set-up, depends on the comparator type configuration (CTC) for each comparator pair. MPU channels could be configured either to check instruction addresses (IA) or operand addresses (OA). IA ranges could be used to define execute permissions. OA ranges could be used to define read and write permissions.

The comparator type for MPU usage could be set to

- CTC=0: both IA ranges define execute permissions,
- CTC=1: both OA ranges define read/write permissions and
- CTC=2: IA range 0 defines execute permissions and OA range 1 defines read/write permissions.

Data value (DT) detection by setting CTC=3 is not possible to use in MPU mode.

Permission configurations exist for read, write and execute for two CPU modes, the supervisor mode and the user mode. Supervisor permissions are valid for SV=1 and user permissions are valid for SV=0.

BIT[23]: SRX1 - Supervisor Read/eXecute permission bit for range 1

Setting valid for CTC == 0 (Instruction address range comparator):

0	Supervisor has no execute permission on address range 1(default)
1	Supervisor has execute permission on address range 1

Setting valid for CTC == 1 or CTC == 2 (Operand address range comparator):

0	Supervisor has no read permission on address range 1 (default)
1	Supervisor has read permission on address range 1

BIT[22]: SW1 - Supervisor Write permission bit for range 1

Setting valid for CTC == 1 or CTC == 2 (Operand address range comparator):

0	Supervisor has no write permission on address range 1 (default)
1	Supervisor has write permission on address range 1

BIT[21]: SRX0 - Supervisor Read/eXecute permission bit for range 0

Setting valid for CTC == 0 or CTC == 2 (Instruction address range comparator):

0	Supervisor has no execute permission on address range 0 (default)
1	Supervisor has execute permission on address range 0

Setting valid for CTC == 1 (Operand address range comparator):

0	Supervisor has no read permission on address range 0 (default)
1	Supervisor has read permission on address range 0

BIT[20]: SW0 - Supervisor Write permission bit for range 0

Setting valid for CTC == 1 (Operand address range comparator):

0	Supervisor has no write permission on address range 0 (default)
1	Supervisor has write permission on address range 0

MB91460 Series**BIT[19]: URX1 - User Read/eXecute permission bit for range 1**

Setting valid for CTC == 0 (Instruction address range comparator):

0	User has no execute permission on address range 1(default)
1	User has execute permission on address range 1

Setting valid for CTC == 1 or CTC == 2 (Operand address range comparator):

0	User has no read permission on address range 1 (default)
1	User has read permission on address range 1

BIT[18]: UW1 - User Write permission bit for range 1

Setting valid for CTC == 1 or CTC == 2 (Operand address range comparator):

0	User has no write permission on address range 1 (default)
1	User has write permission on address range 1

BIT[17]: URX0 - User Read/eXecute permission bit for range 0

Setting valid for CTC == 0 or CTC == 2 (Instruction address range comparator):

0	User has no execute permission on address range 0 (default)
1	User has execute permission on address range 0

Setting valid for CTC == 1 (Operand address range comparator):

0	User has no read permission on address range 0 (default)
1	User has read permission on address range 0

BIT[16]: UW0 - User Write permission bit for range 0

Setting valid for CTC == 1 (Operand address range comparator):

0	User has no write permission on address range 0 (default)
1	User has write permission on address range 0

● Group of Channels, Mode Configuration Register

BIT[15]: MPE - Memory Protection Enable

0	The group of channels operates as debug interface and defines breakpoints (default)
1	The group of channels operates in memory protection mode

Some restrictions apply with the setting of the MPE bit.

MPE=0 (break unit):

- permission registers are don't care (BCRx bits [23:16])

MPE=1 (memory protection unit):

- OBS and OBT should be set to '3' (BCRx bits [11:8], any size and any type)
- CTC should not be set to '3' (BCRx bits [13:12], data value check not supported in this mode)

BIT[14]: COMB - Channel Combination Enable

0	No combination between channels (default)
1	Combination between channels is effective

Depending on the MPE configuration bit the COMB feature has different meaning.

(A) COMB=1 and MPE=0 (break unit, combined operand address and data value break):

The break detection conditions are combined before setting the BIRQ_BD bits and signalise an operand break condition. Setting the COMB bit is required for defining a data value break on a specific operand address. If the COMB bit is set to '1', both conditions, matching operand address (OA) and matching data value (DT), are required to be true. Setting the COMB bit makes only sense in the OA/DT mode, defined by CTC=3.

The AND-combination is effective between channels 3 (OA1) and 1 (DT1) and between channels 2 (OA0) and 0 (DT0). It is assumed that no range operation is defined (ER1=ER0=0).

$BIRQ_BD3 = BIRQ_BD1 = BD3 \ \&\& \ BD1;$

$BIRQ_BD2 = BIRQ_BD0 = BD2 \ \&\& \ BD0;$

If channels 3 and 2 define an operand address range (OA1:OA0) by setting ER1=1 and/or channels 1 and 0 define a data value range (DT1:DT0) by setting ER0=1, the break detection bits of each channel are AND-combined with the combined (logical OR) channels of the opposite range comparator break detection outputs.

$BIRQ_BD3 = BD3 \ \&\& \ (BD1 \ \parallel \ BD0);$

$BIRQ_BD2 = BD2 \ \&\& \ (BD1 \ \parallel \ BD0);$

$BIRQ_BD1 = BD1 \ \&\& \ (BD3 \ \parallel \ BD2);$

$BIRQ_BD0 = BD0 \ \&\& \ (BD3 \ \parallel \ BD2);$

This offers the same interpretation of the BIRQ break detection bits (see [Table 29.3-2 on page 545](#) for coding of match on start point, range or end point) as it would be the case for range detection with COMB=0. BD3 and BD2 hold the coding for the operand address (OA) match, whereas BD1 and BD0 hold the coding for the data value (DT) match. The COMB bit set to '1' ensures that both conditions, the OA match and the DT match must be true to set the appropriate BD bit in the end.

If the COMB bit is set to '0' all break detection bits are passed to the BIRQ register in it's original form. The comparator channels match conditions are independent from each other.

(B) COMB=1 and MPE=1 (memory protection unit, combined rwx permissions on single range):

In memory protection mode the COMB bit has the meaning of combined data read/write and code execute permissions, set for the same address range. The setting is only meaningful for the combination of operand address

MB91460 Series

(OA) comparators on channels 3 and 2 and instruction address (IA) comparators on channels 1 and 2 in the mode CTC=2.

The COMB bit set to '1' causes the IA comparator CMP0 to use the same BADx point definitions as the OA comparator CMP1. Point 3 and Point 2 define the address range for both comparators CMP0 and CMP1. This has the effect that the entry of Point 0/Mask 0 is not allocated for the Point set-up and could be used for masking either one or both comparators. The Point 1 entry is not useable in this case.

If the COMB bit is set to '0' both comparators have independent address configurations. The comparators can either define read/write permissions for data protection or define execute permissions for code protection. Each comparator can define an address region by a range between two points (ER=1) or by one point with a mask (EM=1).

BIT[13:12]: CTC[1:0] - Comparator Type Config

CTC	CMP1	CMP0	Break Function	MPU Function
0	IA	IA	4 instruction break points	2 regions for code protection (x permissions)
1	OA	OA	4 operand break points	2 regions for data protection (rw permissions)
2	OA	IA	2 instruction break points + 2 operand break points	1 region for code protection (x permissions) and 1 region for data protection (rw permissions) or 1 region for combined code and data protection (rwx permissions)
3	OA	DT	2 operand break points + 2 data value breaks, normally with combination	not applicable

One group of channels contains 2 range comparator blocks. Each comparator block can detect a range hit between two points or two independent point hits. The point configuration is stored in dedicated BADx registers for each channel (4 BADx registers for each group of channels).

The comparator type configuration (CTC) controls the input multiplexing of the compare value for each of the two range comparator blocks CMP1 and CMP0. CMP1 combines the break detection channels 3 and 2. The compare value for CMP1 can be assigned either to the instruction address (IA) or to the operand address (OA). CMP0 combines the break detection channels 1 and 0. The compare value for CMP0 can be assigned to the instruction address (IA), to the operand address (OA) or to the data value (DT). The table above defines the input compare values for CMP1 and CMP0, depending on the CTC setting.

In addition a mask for each comparator block could be defined (see the definition of the EM bits later). In this case the BADx register, which contains the mask information, is not available for the point configuration. Thus the usage of the mask feature restricts the number of points or channels, which are available in total.

BIT[11:8]: OBS[1:0], OBT[1:0] - Operand Break Size / Operand Break Type register 1

Datasize			Access type		
OBS1	OBS0		OBT1	OBT0	
0	0	Byte (Default)	0	0	Read (Default)
0	1	Halfword	0	1	Read-Modify-Write
1	0	Word	1	0	Write
1	1	All (Byte, Hword, Word)	1	1	All (Read, RMW, Write)

The operand break size register OBS configures the datasize and the operand break type register OBT

configures the access type if the channel is configured to operand address break or data value break detection.

Setting to 'all' in datasize will cause detection of byte, halfword and word data sizes. Setting to 'all' in access type will cause detection of Read, Read-Modify-Write and Write access types.

● Enable Break Point Register

BIT[7]: EP3 - Enable break Point 3 register

0	Break point 3 register is disabled (default)
1	Break point 3 register is enabled

If EP3 is enabled then the input value of CMP1 will be compared with the point 3 register content (BAD index = 3+group offset, BAD3 for group 0 channel 3, BAD7 for group 1 channel 3, ...).

The input value and the point value is masked if the mask function is enabled by EM1. On a compare match a break exception will be executed. CTC and MPE control the selection of the input value and the type of the break exception.

BIT[6]: EP2 - Enable break Point 2 register

0	Break point 2 register is disabled (default)
1	Break point 2 register is enabled

If EP2 is enabled then the input value of CMP1 will be compared with the point 2 register content (BAD index = 2+group offset, BAD2 for group 0 channel 2, BAD6 for group 1 channel 2, ...).

The input value and the point value is masked if the mask function is enabled by EM1. On a compare match a break exception will be executed. CTC and MPE control the selection of the input value and the type of the break exception.

EP2 controls in addition to enabling and allocating point 2 the selection of the mask register. Point 2 is also the default place for storing the CMP1 mask value. But, if point 2 is enabled, the mask could not be stored there and the mask input of CMP1 switches to point 0 (to the opposite comparator).

BIT[5]: EP1 - Enable break Point 1 register

0	Break point 1 register is disabled (default)
1	Break point 1 register is enabled

If EP1 is enabled then the input value of CMP0 will be compared with the point 1 register content (BAD index = 1+group offset, BAD1 for group 0 channel 1, BAD5 for group 1 channel 1, ...).

The input value and the point value is masked if the mask function is enabled by EM0. On a compare match a break exception will be executed. CTC and MPE control the selection of the input value and the type of the break exception.

MB91460 Series**BIT[4]: EP0 - Enable break Point 0 register**

0	Break point 0 register is disabled (default)
1	Break point 0 register is enabled

If EP0 is enabled then the input value of CMP0 will be compared with the point 0 register content (BAD index = 0+group offset, BAD0 for group 0 channel 0, BAD4 for group 1 channel 0, ...).

The input value and the point value is masked if the mask function is enabled by EM0. On a compare match a break exception will be executed. CTC and MPE control the selection of the input value and the type of the break exception.

EP0 controls in addition to enabling and allocating point 0 the selection of the mask register. Point 0 is also the default place for storing the CMP0 mask value. But, if point 0 is enabled, the mask could not be stored there and the mask input of CMP0 switches to point 2 (to the opposite comparator).

If memory protection is enabled (MPE=1) in conjunction with the combination bit set (COMB=1), the address range is defined by point 3 and point 2 and is valid for both comparators COMB1 and COMB0. So the points 1 and 0 are not required for the range definition of CMP0, independent from the point enable EP0 and EP1, which normally are set in this case. Thus point 0 could be used for storing the mask value for both comparators CMP1 and CMP0 and the exception described in the paragraph above did not apply for this case.

● Enable Mask And Range Register

BIT[3]: EM1 - Enable Mask for CMP1

0	Mask function for CMP1 is disabled (default)
1	Mask function for CMP1 is enabled

If EM1 is enabled the comparator CMP1 matches only these bit positions, which are set to '0' and are not masked by the mask register. All inputs for points and the compare value itself are OR-combined with the value from the mask register. The compare operations point match or range detection are derived based on these OR-masked values.

The selection of the appropriate BADx register (point 2 or 0) for the mask value depends on EP2 and ER1. If at least one of both bits are enabled, the mask usage switches to point 0 due to the allocation of point 2. Otherwise the default mask stored in point 2 applies for CMP1.

BIT[2]: EM0 - Enable Mask for CMP0

0	Mask function for CMP0 is disabled (default)
1	Mask function for CMP0 is enabled

If EM0 is enabled the comparator CMP0 matches only these bit positions, which are set to '0' and are not masked by the mask register. All inputs for points and the compare value itself are OR-combined with the value from the mask register. The compare operations point match or range detection are derived based on these OR-masked values.

The selection of the appropriate BADx register (point 0 or 2) for the mask value depends on EP0 and ER0. If at least one of both bits are enabled, the mask usage switches to point 2 due to the allocation of point 0. Otherwise the default mask stored in point 0 applies for CMP0. If MPE=1 and COMB=1 the mask is taken from point 0, regardless of the setting of EP0 and ER0.

BIT[1]: ER1 - Enable Range for CMP1

0	Range detection CMP1 (channels 2-3) is disabled (default)
1	Range detection CMP1 (channels 2-3) is enabled

If ER1 is enabled then the registers BADx, point 3 and point 2 will be used for range comparison:

Point 2 <= Compare Value <= Point 3.

If a mask is set with EM1 then both point registers will be masked with the mask register content.

Point 3 and Point 2 are taken from BAD[x+3] and BAD[x+2], the mask is stored in Point 0, BAD[x+0].

The 'x' is the group offset and calculates by the group index multiplied with 4.

BIT[0]: ER0 - Enable Range for CMP0

0	Range detection CMP0 (channels 0-1) is disabled (default)
1	Range detection CMP0 (channels 0-1) is enabled

If ER0 is enabled then the registers BADx, point 1 and point 0 will be used for range comparison:

Point 0 <= Compare Value <= Point 1.

If a mask is set with EM0 then both point registers will be masked with the mask register content.

In the special case of MPE=1 together with COMB=1, Point 1 and Point 0 are taken from the opposite channel BAD[x+3] and BAD[x+2] and the mask is stored in Point 0, BAD[x+0]. Otherwise Point 1 and Point 0 are taken from BAD[x+1] and BAD[x+0], the mask is stored in Point 2, BAD[x+2].

The 'x' is the group offset and calculates by the group index multiplied with 4.

■ BAD0...BAD31 (Break Address/Data register0...31)

The BADx registers define 32 break point addresses, data values or mask information for the 8 groups of channels. For each group of channels there are 4 dedicated BAD registers. BAD0, BAD1, BAD2 and BAD3 belong to Group 0, BAD4, BAD5, BAD6 and BAD7 belong to Group 1 and so on. The functionality described below for the registers of group 0 is representative for all the other groups too. The index of the BADx registers has to be incremented by 4 for each of the next group indexes.

BAD0 (BAD4, BAD8, ..., BAD28) [R/W]				
Address	+0	+1	+2	+3
F080 _H	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

This register sets the 32 bit comparison value for break point 0 of CMP0. In range mode (set with ER0) the register value of BAD0 functions as lower address limit. In addition BAD0 could be used as mask register.

In the special case of MPE=1 and COMB=1 BAD0 is not used for the point definition. CMP0 gets its point configuration then from BAD2.

BAD1 (BAD5 BAD9, ..., BAD29) [R/W]				
Address	+0	+1	+2	+3
F084 _H	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

This register sets the 32 bit comparison value for break point 1 of CMP0. In range mode (set with ER0) the register value of BAD1 functions as upper address limit.

In the special case of MPE=1 and COMB=1 BAD1 is not used for the point definition. CMP0 gets its point configuration then from BAD3.

BAD2 (BAD6, BAD10, ..., BAD30) [R/W]				
Address	+0	+1	+2	+3
F088 _H	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

This register sets the 32 bit comparison value for break point 2 of CMP1. In range mode (set with ER1) the register value of BAD2 functions as lower address limit. In addition BAD2 could be used as mask register.

BAD3 (BAD7, BAD11, ..., BAD31) [R/W]				
Address	+0	+1	+2	+3
F08C _H	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

This register sets the 32 bit comparison value for break point 3 of CMP1. In range mode (set with ER1) the register value of BAD3 functions as upper address limit.

29.4. Break Functions

29.4.1 Instruction address break

The instruction address point break is the most basic break that occurs when an instruction is fetched at the address specified by the break address data registers BAD[3:0]. Setting the CTC[1:0] bits of the control register BCR0 to '00' provides this mode. The bits EP[3:0] in BCR0 enable the break points.

Up to 4 instruction breakpoints from channels 0 to 3 can be set. All instruction break events are combined (logical OR) into instruction break exception requests to the CPU.

2 of the break address registers can operate as mask registers (BAD0, BAD2) for masking the instruction address which is being fetched. Mask register BAD0 can be assigned either to BAD1 (same channel) or BAD2/3 (opposite channel), mask register BAD2 can be assigned either to BAD3 or BAD0/1.

Normally Instruction break address and mask information reside in the same channel. So BAD3 contains the instruction break address and BAD2 the address mask information. The channel is enabled with EP3. The same applies for channel BAD1 (address), BAD0 (mask) and EP1 (enable).

But some cases require enabling point 2 (EP2) or the range function (ER1). Then BAD2 holds Instruction Address information and could not carry the address mask. In that cases (when EP2 or ER1 are set) the mask information is taken from the opposite BAD0 register. The same applies for EP0 and ER0 - which enables the use of the opposite BAD2 register for the mask information.

Example:	CTC	00	Type: Instruction Address Break
	EP1	1	Enable break point address BAD1
	EM0	1	Set mask BAD0 for break address BAD1
	BAD1	0x12345678	Set break address
	BAD0	0x00000FFF	Set break mask

Break occurs at 0x12345000 to 0x12345FFF

On break at BAD[3:0] the respective flags BD[3:0] in the break interrupt register BIRQ will be set to '1'. They have to be reset by software in the instruction break routine.

Channels 0 and 1 (BAD0, BAD1) can be set up to function as address range match. Setting the ER0 bit of the control register BCR0 to '1' provides this mode. BAD0 is the lower address and BAD1 is the upper address for address comparison. In this mode the mask register BAD2 will mask both channels 0 and 1, if the mask feature is enabled by EM0 = '1'.

Alternatively channels 2 and 3 (BAD2, BAD3) can be set up to function as address range match. Setting the ER1 bit of the control register BCR0 to '1' provides this mode. BAD2 is the lower address and BAD3 is the upper address for address comparison. In this mode the mask register BAD0 will mask both channels 2 and 3, if the mask feature is enabled by EM1 = '1'.

Example:	CTC	00	Type: Instruction Address Break
	EP0	1	Enable break point on BAD0
	EP1	1	Enable break point on BAD1
	ER0	1	Enable address range function on BAD0, BAD1
	EM0	1	Enable address mask function on BAD0, BAD1
	BAD0	0x12345200	Set lower break address
	BAD1	0x12345300	Set upper break address
	BAD2	0xF0000000	Set break mask

Break occurs at 0x02345200 to 0x02345300, or
at 0x12345200 to 0x12345300, or

MB91460 Series

at 0x22345200 to 0x22345300, etc.

The resulting setting of the BD[1:0] status bits indicates the point, respective the area in which the break has occurred.

Table 29.4-1 Instruction Break Detection Status Bits (BD)

BD1	BD0	
0	1	Match on point (instruction address == 0x12345200), or Match on point (instruction address == 0x22345200), etc
1	0	Match on point (instruction address == 0x12345300), or Match on point (instruction address == 0x22345300), etc
1	1	Match on range (0x12345200 < instruction address < 0x12345300), or Match on range (0x22345200 < instruction address < 0x22345300), etc

In the instruction address break mode the following important point has to be considered:

To precisely determine the instruction address where a break occurs, use the PC value saved on the stack during entry to the instruction break interrupt service routine.

29.4.2 Operand address break

The operand break function causes a break for the data access address which can be specified by the operand address break registers BAD[3:0]. Setting the CTC[1:0] bits of the control register BCR0 to '01' provides this mode. The bits EP[3:0] in BCR0 enable the break points.

Up to 4 breakpoints from channels 0 to 3 can be set. All operand break events are combined (logical OR) into a operand break exception interrupt request to the CPU.

For the address mask function the same applies, what is stated in section [29.4.1 Instruction address break \(Page No.556\)](#) for the Instruction Address Break.

Example:	CTC	01	Type: Operand Address Break
	EP1	1	Enable break point address BAD1
	EM0	1	Set mask BAD0 for break address BAD1
	BAD1	0x12345678	Set break address
	BAD0	0x00000FFF	Set break mask

Break occurs at 0x12345000 to 0x12345FFF

On break at BAD[3:0] the respective flags BD[3:0] in the break interrupt register BIRQ will be set to '1'. They have to be reset by software in the operand break exception routine.

Channels 0 and 1 (BAD0, BAD1) can be set up to function as address range match. Setting the ER0 bit of the control register BCR0 to '1' provides this mode. BAD0 is the lower address and BAD1 is the upper address for address comparison. In this mode the mask register BAD2 will mask both channels 0 and 1, if the mask feature is enabled by EM0 = '1'.

Alternatively channels 2 and 3 (BAD2, BAD3) can be set up to function as address range match. Setting the ER1 bit of the control register BCR0 to '1' provides this mode. BAD2 is the lower address and BAD3 is the upper address for address comparison. In this mode the mask register BAD0 will mask both channels 2 and 3, if the mask feature is enabled by EM1 = '1'.

Example:	CTC	01	Type: Operand Address Break
	EP0	1	Enable break point on BAD0

EP1	1	Enable break point on BAD1
ER0	1	Enable address range function on BAD0, BAD1
EM0	1	Enable address mask function on BAD0, BAD1
BAD0	0x12345200	Set lower break address
BAD1	0x12345300	Set upper break address
BAD2	0xF0000000	Set break mask

Break occurs at 0x02345200 to 0x02345300, or
 at 0x12345200 to 0x12345300, or
 at 0x22345200 to 0x22345300, etc.

The resulting setting of the BD[1:0] status bits indicates the point, respective the area in which the break has occurred.

Table 29.4-2 Operand Break Detection Status Bits (BD)

BD1	BD0	
0	1	Match on point (operand address == 0x12345200), or Match on point (operand address == 0x22345200), etc
1	0	Match on point (operand address == 0x12345300), or Match on point (operand address == 0x22345300), etc
1	1	Match on range (0x12345200 < operand address < 0x12345300), or Match on range (0x22345200 < operand address < 0x22345300), etc

The access data length and read/write break attributes can also be specified by the control register BCR0, bits OBS[1:0] and OBT[1:0]. When the mask function is disabled by setting EM1 = EM0 = 0 (all bits effective), the relationship between breakpoint setting, and break by access address is shown below:

Table 29.4-3 Operand size and operand address relations

Access data length	Access address	Address set in BOAC			
		4n + 0	4n + 1	4n + 2	4n + 3
8 bit	4n + 0	Hit	-	-	-
	4n + 1	-	Hit	-	-
	4n + 2	-	-	Hit	-
	4n + 3	-	-	-	Hit
16 bit	4n + 0	Hit	Hit	-	-
	4n + 1	Hit	Hit	-	-
	4n + 2	-	-	Hit	Hit
	4n + 3	-	-	Hit	Hit

Table 29.4-3 Operand size and operand address relations

Access data length	Access address	Address set in BOAC			
		4n + 0	4n + 1	4n + 2	4n + 3
32 bit	4n + 0	Hit	Hit	Hit	Hit
	4n + 1	Hit	Hit	Hit	Hit
	4n + 2	Hit	Hit	Hit	Hit
	4n + 3	Hit	Hit	Hit	Hit

In Operand address break mode the Operand Address, causing the break is captured in the BOAC register. Additional BIAC holds the instruction address of the instruction, which was executed one cycle before the break causing data operation. This is normally the instruction, which has caused the data transfer.

In the operand address break mode the following important points have to be considered:

- 1) In the FR family architecture, if data access is performed with misalignment, the lower address bit 0 will be ignored for halfword and the lower address bits 0 and 1 for word access. The mask register could be programmed accordingly.
- 2) The EDSU operand break does not always occur immediately after completion of execution of the instruction causing the break event.
- 3) Please see also information at chapter [29.4.4 Using operand with data break \(Page No.560\)](#).

29.4.3 Data value break

The data value break causes a break if specified data is read or written at a data access to an address specified by the CPU. The data can be specified by the data value break registers BAD0 and BAD1. Setting the CTC[1:0] bits of the control register BCR0 to '11' provides this mode. The bits EP0 and EP1 in BCR0 enable the break condition.

Up to 2 break points from channels 0 to 1 can be set. All data value break events are combined (logical OR) into a operand break exception to the CPU.

1 mask register (BAD0) is available for masking the data value (stored in BAD1) and 1 mask register (BAD2) is available for masking the operand address (BAD3) which is being accessed. Mask registers BAD2 and BAD0 can be enabled with EM1 and EM0.

The data on which a break should be executed must be masked by a data-mask on the bus, requiring 32-bit setting considering the address and data length (see table below). This is required due to the byte position of the operand is dependent from the operand address. The setting of data length of the control register BCR0 OBS[1:0] could be configured to all ignored. The data length is controllable by mask setting to the BAD0 register implicitly.

On break at BAD[1:0] the respective flags BD[1:0] in the break interrupt register BIRQ will be set to '1'. They have to be reset by software in the operand break exception routine.

In Operand data value break mode the Operand Address, causing the break is captured in the BOAC register. Additional BIAC holds the instruction address of the instruction, which was executed one cycle before the break causing data operation. This is normally the instruction, which has caused the data transfer.

In the data value break mode the following important points have to be considered:

- 1) The data value break is also executed for matching DMA transfers. This could lead to unexpected behaviour due to parallel processes. The filter bits FDMA and FCPU could be set for dedicated investigations.

2) The EDSU data break does not always occur immediately after completion of execution of the instruction causing the break event.

3) Please see also information at chapter [29.4.4 Using operand with data break \(Page No.560\)](#)

Table 29.4-4 Data size and data address relations

Access data length	Address set to BAD1/0	MASK set to BAD0	Position of valid data in BAD1/0 (indicated by *)	Remarks
8 bit	4n + 0	0x00FFFFFF	**__ ____	
	4n + 1	0xFF00FFFF	__** ____	
	4n + 2	0xFFFF00FF	____ **__	
	4n + 3	0xFFFFF000	____ __**	
16 bit	4n + 0	0x0000FFFF	**** ____	Possibly intended to use address mask in BAD1 for address bit 0
	4n + 1	0x0000FFFF	**** ____	
	4n + 2	0xFFFF0000	____ ****	
	4n + 3	0xFFFF0000	____ ****	
32 bit	4n + 0	0x00000000	**** ****	Possibly intended to use address mask in BAD1 for address bits 1 and 0; Data mask not required, two channels could be used
	4n + 1	0x00000000	**** ****	
	4n + 2	0x00000000	**** ****	
	4n + 3	0x00000000	**** ****	

Notes:

1) The mask values for the BAD0 register in the table are a minimum set of bits. Setting more masking bits permits masking bits not needed to be compared with transfer data.

2) "Position of valid data in BAD1, BAD0" provides an 8-bit hexadecimal image for MSB on the left and LSB on the right. Data at bit positions indicated by * in the BAD1, BAD0 registers is compared with data on the data bus, according to the access data length and access address.

29.4.4 Using operand with data break

Using operand address with data value break together is enabled with setting both EP3 and EP1, and/or both EP2 and EP0 together with setting the bit COMB = '1' for the data value break mode set with CTC = '11'.

In other words: a break in channel 0 will occur at a match on operand address in BAD2 and a match on data value in BAD0. A break in channel 1 will occur at a match on operand address in BAD3 and a match on data value in BAD1. It is not possible to mix them vice versa.

On break both BD0 and BD2, respective BD1 and BD3 are set. They have to be reset by software in the

MB91460 Series

operand break exception routine.

Table 29.4-5 Operand address and data value break combinations

EP3/2	EP1/0	COMB	Function
0	0	0	No break detection
0	1	0	Independent data break (match value on any operand address)
1	0	0	Independent Operand break (match operand address)
1	1	0	Independent Data break and Operand break
0	0	1	No break detection
0	1	1	No break detection
1	0	1	No break detection
1	1	1	Data value break (match both operand address and value)

29.4.5 Memory Protection

Due to the availability of address range comparators for the operand and instruction addresses the wish is obvious, to use the same comparator hardware as a memory protection unit (MPU).

Following table list the possible type configurations and its feasibility to be used for memory protection. The number of break points and MPU channels is valid for 8 comparator groups implemented.

Table 29.4-6 Comparator Type Configuration

CTC	CMP1 Input	CMP0 Input	Max. Break Points (MPE=0)	Max. MPU Channels (MPE=1)
00	IA	IA	32 Instruction breaks	16 ranges with execute permissions
01	OA	OA	32 Operand breaks	16 ranges with read/write permissions
10	OA	IA	16 + 16 IA/OA breaks	8 ranges with read/write and execute permissions or 8+8 independent ranges
11	OA	DT	16 data value breaks	-

Additional to the given hardware there were made some extensions to provide the user with a more likely configuration of read, write and execute permissions instead of the more bus applicable definition of the operand break size and type definitions OBS/OBT, including read-modify-write. With the introduction of the Supervisor mode a definition of User and Supervisor permissions is possible.

Permissions can be set for the comparator channel CMP1 and CMP0 separately, indicated by the symbol index.

Table 29.4-7 Meaning of the permission config bits

Symbol	Data Mode (OA)	Instruction Mode (IA)
SRX[1:0]	Supervisor Read permission	Supervisor eXecute permission
SW[1:0]	Supervisor Write permission	-

Table 29.4-7 Meaning of the permission config bits

Symbol	Data Mode (OA)	Instruction Mode (IA)
URX[1:0]	User Read permission	User eXecute permission
UW[1:0]	User Write permission	-

At each time an instruction is executed or an operand is accessed, the actual valid permissions were evaluated. This evaluation is divided into operand access (OA-based) and code execution (IA-based).

For each part the highest priority region hit is searched for. The highest channel number has the highest priority (strict priority scheme). If a channel hit was found, the permissions defined for this channel will apply. If no channel hit was detected, the default permissions apply.

After the actual permissions are evaluated (valid for actual data access, if any, and actual instruction) the permissions were checked. If the execute permission is not set or if the read or write permissions do not fit to the type of the actual access, a protection violation will be indicated. This causes a CPU trap to the memory protection violation MPUPV handler routine. The CPU switches directly to Supervisor mode in this case.

The config register space of the EDSU is protected against random access in User mode. Only in Supervisor mode or Emulation mode the register file enables write access. For configuration a system interrupt INT #5 was defined, which switches in Supervisor mode (SV bit remains set during the execution of the INT #5-ISR). Except debugger interrupts by the emulator and NMI the Supervisor ISR is not interruptible.

Exceptions caused by the memory protection and the break unit for debugging are separated. In that way the memory protection functionality can be debugged itself.

29.4.6 Break Factors

Summary of the internal break factors and the executed events:

Break on instruction address	->	Causes Instruction Break
Break on operand address	->	Causes Operand Break
Break on data value	->	Causes Operand Break
Resource Interrupt (BREAK)	->	Causes Tool NMI
Step Trace Trap	->	Causes Step Trace Trap
Execution of the INTE instr.	->	Causes INTE
Execution of INT #5	->	CPU Supervisor Mode
Memory protection exception	->	Causes MPUPV Trap

Break factors and corresponding interrupt numbers and vectors:

Table 29.4-8 Interrupt numbers and vectors of break factors

Interrupt	Interrupt number		Interrupt level		Interrupt vector	
	Decimal	Hexa-decimal	Setting Register	Register address	Offset	Default Vector address
CPU supervisor mode (INT #5 instruction)	5	05	-	-	0x3E8	0x000FFFE8
Memory protection exception	6	06	-	-	0x3E4	0x000FFFE4
INTE instruction	9	09	-	-	0x3D8	0x000FFFD8

MB91460 Series**Table 29.4-8 Interrupt numbers and vectors of break factors**

Interrupt	Interrupt number		Interrupt level		Interrupt vector	
	Decimal	Hexa-decimal	Setting Register	Register address	Offset	Default Vector address
Instruction break exception	10	0A	-	-	0x3D4	0x000FFFD4
Operand break exception	11	0B	-	-	0x3D0	0x000FFFD0
Step trace trap	12	0C	-	-	0x3CC	0x000FFFC8
NMI interrupt (tool)	13	0D	-	-	0x3C8	0x000FFFC8

Chapter 30 I/O Ports

30.1. I/O Port Functions MB91V460A

For enabling the resource functions, please refer to section [30.4.4 Port Function Register Setup \(Page No.598\)](#).

Table 30.1-1 I/O port functions MB91V460A

Pin Name	I/O Signal	Circuit Type	Function
INITX	INITX	TC02_0	Reset input, low active
RSTX	RSTX	TC01_0	Reset input, low active
HSTX	HSTX	TC00_0	Hardware Standby input, low active
NMIX	NMIX	TC01_0	Non-maskable Interrupt input, low active
MD2	MD2	TC02_1	Mode Pin 2
MD1	MD1	TC02_1	Mode Pin 1
MD0	MD0	TC02_1	Mode Pin 0
X0	X0	TO00_0	Main Oscillation input
X1	X1	TO00_1	Main Oscillation output
X0A	X0A	TO01_0	Sub Oscillation input
X1A	X1A	TO01_1	Sub Oscillation output
ALARM0	ALARM0	TA02_0	ALARM comparator input channel 0
ALARM1	ALARM1	TA02_0	ALARM comparator input channel 1
EBREAKX	EBREAKX	TC01_0	EDSU BREAK input, low active
MONCLK	MONCLK	TC10_0	Clock Monitor Output
Port 00			
P00_7	P00_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D31		I/O pin for bit 31 of the external data bus. This function is enabled when the external bus is enabled.
P00_6	P00_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D30		I/O pin for bit 30 of the external data bus. This function is enabled when the external bus is enabled.
P00_5	P00_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D29		I/O pin for bit 29 of the external data bus. This function is enabled when the external bus is enabled.
P00_4	P00_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D28		I/O pin for bit 28 of the external data bus. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P00_3	P00_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D27		I/O pin for bit 27 of the external data bus. This function is enabled when the external bus is enabled.
P00_2	P00_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D26		I/O pin for bit 26 of the external data bus. This function is enabled when the external bus is enabled.
P00_1	P00_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D25		I/O pin for bit 25 of the external data bus. This function is enabled when the external bus is enabled.
P00_0	P00_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D24		I/O pin for bit 24 of the external data bus. This function is enabled when the external bus is enabled.
Port 01			
P01_7	P01_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D23		I/O pin for bit 23 of the external data bus. This function is enabled when the external bus is enabled.
P01_6	P01_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D22		I/O pin for bit 22 of the external data bus. This function is enabled when the external bus is enabled.
P01_5	P01_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D21		I/O pin for bit 21 of the external data bus. This function is enabled when the external bus is enabled.
P01_4	P01_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D20		I/O pin for bit 20 of the external data bus. This function is enabled when the external bus is enabled.
P01_3	P01_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D19		I/O pin for bit 19 of the external data bus. This function is enabled when the external bus is enabled.
P01_2	P01_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D18		I/O pin for bit 18 of the external data bus. This function is enabled when the external bus is enabled.
P01_1	P01_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D17		I/O pin for bit 17 of the external data bus. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P01_0	P01_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D16		I/O pin for bit 16 of the external data bus. This function is enabled when the external bus is enabled.
Port 02			
P02_7	P02_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D15		I/O pin for bit 15 of the external data bus. This function is enabled when the external bus is enabled.
P02_6	P02_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D14		I/O pin for bit 14 of the external data bus. This function is enabled when the external bus is enabled.
P02_5	P02_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D13		I/O pin for bit 13 of the external data bus. This function is enabled when the external bus is enabled.
P02_4	P02_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D12		I/O pin for bit 12 of the external data bus. This function is enabled when the external bus is enabled.
P02_3	P02_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D11		I/O pin for bit 11 of the external data bus. This function is enabled when the external bus is enabled.
P02_2	P02_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D10		I/O pin for bit 10 of the external data bus. This function is enabled when the external bus is enabled.
P02_1	P02_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D9		I/O pin for bit 9 of the external data bus. This function is enabled when the external bus is enabled.
P02_0	P02_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D8		I/O pin for bit 8 of the external data bus. This function is enabled when the external bus is enabled.
Port 03			
P03_7	P03_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D7		I/O pin for bit 7 of the external data bus. This function is enabled when the external bus is enabled.
P03_6	P03_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D6		I/O pin for bit 6 of the external data bus. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P03_5	P03_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D5		I/O pin for bit 5 of the external data bus. This function is enabled when the external bus is enabled.
P03_4	P03_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D4		I/O pin for bit 4 of the external data bus. This function is enabled when the external bus is enabled.
P03_3	P03_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D3		I/O pin for bit 3 of the external data bus. This function is enabled when the external bus is enabled.
P03_2	P03_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D2		I/O pin for bit 2 of the external data bus. This function is enabled when the external bus is enabled.
P03_1	P03_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D1		I/O pin for bit 1 of the external data bus. This function is enabled when the external bus is enabled.
P03_0	P03_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	D0		I/O pin for bit 0 of the external data bus. This function is enabled when the external bus is enabled.
Port 04			
P04_7	P04_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A31		I/O pin for bit 31 of the external address bus. This function is enabled when the external bus is enabled.
P04_6	P04_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A30		I/O pin for bit 30 of the external address bus. This function is enabled when the external bus is enabled.
P04_5	P04_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A29		I/O pin for bit 29 of the external address bus. This function is enabled when the external bus is enabled.
P04_4	P04_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A28		I/O pin for bit 28 of the external address bus. This function is enabled when the external bus is enabled.
P04_3	P04_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A27		I/O pin for bit 27 of the external address bus. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P04_2	P04_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A26		I/O pin for bit 26 of the external address bus. This function is enabled when the external bus is enabled.
P04_1	P04_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A25		I/O pin for bit 25 of the external address bus. This function is enabled when the external bus is enabled.
P04_0	P04_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A24		I/O pin for bit 24 of the external address bus. This function is enabled when the external bus is enabled.
Port 05			
P05_7	P05_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A23		I/O pin for bit 23 of the external address bus. This function is enabled when the external bus is enabled.
P05_6	P05_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A22		I/O pin for bit 22 of the external address bus. This function is enabled when the external bus is enabled.
P05_5	P05_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A21		I/O pin for bit 21 of the external address bus. This function is enabled when the external bus is enabled.
P05_4	P05_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A20		I/O pin for bit 20 of the external address bus. This function is enabled when the external bus is enabled.
P05_3	P05_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A19		I/O pin for bit 19 of the external address bus. This function is enabled when the external bus is enabled.
P05_2	P05_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A18		I/O pin for bit 18 of the external address bus. This function is enabled when the external bus is enabled.
P05_1	P05_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A17		I/O pin for bit 17 of the external address bus. This function is enabled when the external bus is enabled.
P05_0	P05_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A16		I/O pin for bit 16 of the external address bus. This function is enabled when the external bus is enabled.
Port 06			

Pin Name	I/O Signal	Circuit Type	Function
P06_7	P06_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A15		I/O pin for bit 15 of the external address bus. This function is enabled when the external bus is enabled.
P06_6	P06_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A14		I/O pin for bit 14 of the external address bus. This function is enabled when the external bus is enabled.
P06_5	P06_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A13		I/O pin for bit 13 of the external address bus. This function is enabled when the external bus is enabled.
P06_4	P06_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A12		I/O pin for bit 12 of the external address bus. This function is enabled when the external bus is enabled.
P06_3	P06_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A11		I/O pin for bit 11 of the external address bus. This function is enabled when the external bus is enabled.
P06_2	P06_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A10		I/O pin for bit 10 of the external address bus. This function is enabled when the external bus is enabled.
P06_1	P06_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A9		I/O pin for bit 9 of the external address bus. This function is enabled when the external bus is enabled.
P06_0	P06_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A8		I/O pin for bit 8 of the external address bus. This function is enabled when the external bus is enabled.
Port 07			
P07_7	P07_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A7		I/O pin for bit 7 of the external address bus. This function is enabled when the external bus is enabled.
P07_6	P07_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A6		I/O pin for bit 6 of the external address bus. This function is enabled when the external bus is enabled.
P07_5	P07_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A5		I/O pin for bit 5 of the external address bus. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P07_4	P07_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A4		I/O pin for bit 4 of the external address bus. This function is enabled when the external bus is enabled.
P07_3	P07_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A3		I/O pin for bit 3 of the external address bus. This function is enabled when the external bus is enabled.
P07_2	P07_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A2		I/O pin for bit 2 of the external address bus. This function is enabled when the external bus is enabled.
P07_1	P07_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A1		I/O pin for bit 1 of the external address bus. This function is enabled when the external bus is enabled.
P07_0	P07_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	A0		I/O pin for bit 0 of the external address bus. This function is enabled when the external bus is enabled.
Port 08			
P08_7	P08_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	RDY		Input pin for external wait (if RDY enabled for the corresponding CS area). This function is enabled when the external bus is enabled.
P08_6	P08_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	BRQ		Input pin for external bus request (if sharing enabled for the corresponding CS area). This function is enabled when the external bus is enabled.
P08_5	P08_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	BGRNTX		Output pin for external bus granted (if sharing enabled for the corresponding CS area). This function is enabled when the external bus is enabled.
P08_4	P08_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	RDX		Output pin for external bus read strobe. This function is enabled when the external bus is enabled.
P08_3	P08_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	WRX3		Output pin for external bus write strobe. This function is enabled when the external bus is enabled.
P08_2	P08_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	WRX2		Output pin for external bus write strobe. This function is enabled when the external bus is enabled.

Pin Name	I/O Signal	Circuit Type	Function
P08_1	P08_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	WRX1		Output pin for external bus write strobe. This function is enabled when the external bus is enabled.
P08_0	P08_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	WRX0		Output pin for external bus write strobe. This function is enabled when the external bus is enabled.
Port 09			
P09_7	P09_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX7		Output pin for external bus chip select area 7. This function is enabled when the external bus is enabled.
P09_6	P09_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX6		Output pin for external bus chip select area 6. This function is enabled when the external bus is enabled.
P09_5	P09_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX5		Output pin for external bus chip select area 5. This function is enabled when the external bus is enabled.
P09_4	P09_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX4		Output pin for external bus chip select area 4. This function is enabled when the external bus is enabled.
P09_3	P09_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX3		Output pin for external bus chip select area 3. This function is enabled when the external bus is enabled.
P09_2	P09_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX2		Output pin for external bus chip select area 2. This function is enabled when the external bus is enabled.
P09_1	P09_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX1		Output pin for external bus chip select area 1. This function is enabled when the external bus is enabled.
P09_0	P09_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	CSX0		Output pin for external bus chip select area 0. This function is enabled when the external bus is enabled.
Port 10			
P10_7	P10_7	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.

Pin Name	I/O Signal	Circuit Type	Function
P10_6	P10_6	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	MCLKE		Output pin for external bus memory clock enable. This function is enabled when the external bus is enabled.
P10_5	P10_5	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	MCLKI		Input pin for external bus memory clock. This function is enabled when the external bus is enabled.
	MCLKIX		Input pin for external bus memory clock (inverted input). This function is enabled when the external bus is enabled.
P10_4	P10_4	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	MCLKO		Output pin for external bus memory clock. This function is enabled when the external bus is enabled.
	MCLKOX		Output pin for external bus memory clock (inverted output). This function is enabled when the external bus is enabled.
P10_3	P10_3	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	WEX		Output pin for external bus write strobe. This function is enabled when the external bus is enabled.
P10_2	P10_2	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	BAAX		Output pin for external bus burst access. This function is enabled when the external bus is enabled.
P10_1	P10_1	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	ASX		Output pin for external bus address strobe. This function is enabled when the external bus is enabled.
P10_0	P10_0	TP04_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	SYSCLK		Output pin for external bus clock. This function is enabled when the external bus is enabled.
	SYSCLKX		Output pin for external bus clock (inverted output). This function is enabled when the external bus is enabled.
Port 11			
P11_7	P11_7	TP04_0	General purpose I/O.
P11_6	P11_6	TP04_0	General purpose I/O.
P11_5	P11_5	TP04_0	General purpose I/O.
P11_4	P11_4	TP04_0	General purpose I/O.
P11_3	P11_3	TP04_0	General purpose I/O.
P11_2	P11_2	TP04_0	General purpose I/O.
P11_1	P11_1	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	IOWRX		Output pin for DMA memory to I/O fly-by transfer.

Pin Name	I/O Signal	Circuit Type	Function
P11_0	P11_0	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	IORDX		Output pin for DMA I/O to memory fly-by transfer.
Port 12			
P12_7	P12_7	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOP3		Output pin for DMA end of transfer.
P12_6	P12_6	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOTX3		Input pin for DMA transfer stop request.
	DEOP3		Output pin for DMA end of transfer.
P12_5	P12_5	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DACKX3		Output pin for DMA transfer request acknowledgement.
P12_4	P12_4	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DREQ3		Input pin for DMA transfer request.
P12_3	P12_3	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOP2		Output pin for DMA end of transfer.
P12_2	P12_2	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOTX2		Input pin for DMA transfer stop request.
	DEOP2		Output pin for DMA end of transfer.
P12_1	P12_1	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DACKX2		Output pin for DMA transfer request acknowledgement.
P12_0	P12_0	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DREQ2		Input pin for DMA transfer request.
Port 13			
P13_7	P13_7	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOP1		Output pin for DMA end of transfer.
P13_6	P13_6	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOTX1		Input pin for DMA transfer stop request.
	DEOP1		Output pin for DMA end of transfer.
P13_5	P13_5	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DACKX1		Output pin for DMA transfer request acknowledgement.
P13_4	P13_4	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DREQ1		Input pin for DMA transfer request.

Pin Name	I/O Signal	Circuit Type	Function
P13_3	P13_3	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOP0		Output pin for DMA end of transfer.
P13_2	P13_2	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DEOTX0		Input pin for DMA transfer stop request.
	DEOP0		Output pin for DMA end of transfer.
P13_1	P13_1	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DACKX0		Output pin for DMA transfer request acknowledgement.
P13_0	P13_0	TP04_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	DREQ0		Input pin for DMA transfer request.
Port 14			
P14_7	P14_7	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU7		Data sample input pin for input capture ICU 7.
	TIN7		Event input pin for the reload timer RLT 7.
	TTG15/7		Event input pin for the programmable pulse generators PPG 15 and PPG 7.
P14_6	P14_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU6		Data sample input pin for input capture ICU 6.
	TIN6		Event input pin for the reload timer RLT 6.
	TTG14/6		Event input pin for the programmable pulse generators PPG 14 and PPG 6.
P14_5	P14_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU5		Data sample input pin for input capture ICU 5.
	TIN5		Event input pin for the reload timer RLT 5.
	TTG13/5		Event input pin for the programmable pulse generators PPG 13 and PPG 5.
P14_4	P14_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU4		Data sample input pin for input capture ICU 4.
	TIN4		Event input pin for the reload timer RLT 4.
	TTG12/4		Event input pin for the programmable pulse generators PPG 12 and PPG 4.
P14_3	P14_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU3		Data sample input pin for input capture ICU 3.
	TIN3		Event input pin for the reload timer RLT 3.
	TTG11/3		Event input pin for the programmable pulse generators PPG 11 and PPG 3.

Pin Name	I/O Signal	Circuit Type	Function
P14_2	P14_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU2		Data sample input pin for input capture ICU 2.
	TIN2		Event input pin for the reload timer RLT 2.
	TTG10/2		Event input pin for the programmable pulse generators PPG 10 and PPG 2.
P14_1	P14_1	TP00_0	General purpose I/O.
	ICU1		Data sample input pin for input capture ICU 1.
	TIN1		Event input pin for the reload timer RLT 1.
	TTG9/1		Event input pin for the programmable pulse generators PPG 9 and PPG 1.
P14_0	P14_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	ICU0		Data sample input pin for input capture ICU 0.
	TIN0		Event input pin for the reload timer RLT 0.
	TTG8/0		Event input pin for the programmable pulse generators PPG 8 and PPG 0.
Port 15			
P15_7	P15_7	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU7		Waveform output pin for output compare OCU 7.
	TOT7		Output pin for the reload timer RLT 7.
P15_6	P15_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU6		Waveform output pin for output compare OCU 6.
	TOT6		Output pin for the reload timer RLT 6.
P15_5	P15_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU5		Waveform output pin for output compare OCU 5.
	TOT5		Output pin for the reload timer RLT 5.
P15_4	P15_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU4		Waveform output pin for output compare OCU 4.
	TOT4		Output pin for the reload timer RLT 4.
P15_3	P15_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU3		Waveform output pin for output compare OCU 3.
	TOT3		Output pin for the reload timer RLT 3.
P15_2	P15_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU2		Waveform output pin for output compare OCU 2.
	TOT2		Output pin for the reload timer RLT 2.
P15_1	P15_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU1		Waveform output pin for output compare OCU 1.
	TOT1		Output pin for the reload timer RLT 1.

Pin Name	I/O Signal	Circuit Type	Function
P15_0	P15_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	OCU0		Waveform output pin for output compare OCU 0.
	TOT0		Output pin for the reload timer RLT 0.
Port 16			
P16_7	P16_7	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG15		Waveform output pin for programmable pulse generator PPG 15.
	ATGX		A/D converter external trigger input.
P16_6	P16_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG14		Waveform output pin for programmable pulse generator PPG 14.
	PFM		Waveform output pin for pulse frequency modulator PFM.
P16_5	P16_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG13		Waveform output pin for programmable pulse generator PPG 13.
	SGO		Waveform output pin for sound generator SG.
P16_4	P16_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG12		Waveform output pin for programmable pulse generator PPG 12.
	SGA		Amplitude output pin for sound generator SG.
P16_3	P16_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG11		Waveform output pin for programmable pulse generator PPG 11.
P16_2	P16_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG10		Waveform output pin for programmable pulse generator PPG 10.
P16_1	P16_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG9		Waveform output pin for programmable pulse generator PPG 9.
P16_0	P16_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG8		Waveform output pin for programmable pulse generator PPG 8.
Port 17			
P17_7	P17_7	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG7		Waveform output pin for programmable pulse generator PPG 7.
P17_6	P17_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG6		Waveform output pin for programmable pulse generator PPG 6.
P17_5	P17_5	TP00_0	General purpose I/O. This function is enabled in the single-chip mode or by setting the corresponding PFR to '0'.
	PPG5		Waveform output pin for programmable pulse generator PPG 5.

Pin Name	I/O Signal	Circuit Type	Function
P17_4	P17_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG4		Waveform output pin for programmable pulse generator PPG 4.
P17_3	P17_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG3		Waveform output pin for programmable pulse generator PPG 3.
P17_2	P17_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG2		Waveform output pin for programmable pulse generator PPG 2.
P17_1	P17_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG1		Waveform output pin for programmable pulse generator PPG 1.
P17_0	P17_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	PPG0		Waveform output pin for programmable pulse generator PPG 0.
Port 18			
P18_7	P18_7	TP00_0	General purpose I/O.
P18_6	P18_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK7		Clock I/O pin for LIN-USART 7.
	ZIN3		8-bit reset input pin of the Up-/Down Counter UDC 2/3.
	CK7		Input for the 16-bit I/O Timer FRT 7.
P18_5	P18_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT7		Serial data output pin for LIN-USART 7.
	BIN3		8-bit down-count input pin of the Up-/Down Counter UDC 2/3.
P18_4	P18_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN7		Serial data input pin for LIN-USART 7.
	AIN3		8-bit up-count input pin of the Up-/Down Counter UDC 2/3.
P18_3	P18_3	TP00_0	General purpose I/O.
P18_2	P18_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK6		Clock I/O pin for LIN-USART 6.
	ZIN2		8/16-bit reset input pin of the Up-/Down Counter UDC 2/3.
	CK6		Input for the 16-bit I/O Timer FRT 6.
P18_1	P18_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT6		Serial data output pin for LIN-USART 6.
	BIN2		8/16-bit down-count input pin of the Up-/Down Counter UDC 2/3.

Pin Name	I/O Signal	Circuit Type	Function
P18_0	P18_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN6		Serial data input pin for LIN-USART 6.
	AIN2		8/16-bit up-count input pin of the Up-/Down Counter UDC 2/3.
Port 19			
P19_7	P19_7	TP00_0	General purpose I/O.
P19_6	P19_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK5		Clock I/O pin for LIN-USART 5.
	CK5		Input for the 16-bit I/O Timer FRT 5.
P19_5	P19_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT5		Serial data output pin for LIN-USART 5.
P19_4	P19_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN5		Serial data input pin for LIN-USART 5.
P19_3	P19_3	TP00_0	General purpose I/O.
P19_2	P19_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK4		Clock I/O pin for LIN-USART 4.
	CK4		Input for the 16-bit I/O Timer FRT 4.
P19_1	P19_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT4		Serial data output pin for LIN-USART 4.
P19_0	P19_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN4		Serial data input pin for LIN-USART 4.
Port 20			
P20_7	P20_7	TP00_0	General purpose I/O.
P20_6	P20_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK3		Clock I/O pin for LIN-USART 3.
	ZIN1		8-bit reset input pin of the Up-/Down Counter UDC 0/1.
	CK3		Input for the 16-bit I/O Timer FRT 3.
P20_5	P20_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT3		Serial data output pin for LIN-USART 3.
	BIN1		8-bit down-count input pin of the Up-/Down Counter UDC 0/1.
P20_4	P20_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN3		Serial data input pin for LIN-USART 3.
	AIN1		8-bit up-count input pin of the Up-/Down Counter UDC 0/1.

Pin Name	I/O Signal	Circuit Type	Function
P20_3	P20_3	TP00_0	General purpose I/O.
P20_2	P20_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK2		Clock I/O pin for LIN-USART 2.
	ZIN0		8/16-bit reset input pin of the Up-/Down Counter UDC 0/1.
	CK2		Input for the 16-bit I/O Timer FRT 2.
P20_1	P20_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT2		Serial data output pin for LIN-USART 2.
	BIN0		8/16-bit down-count input pin of the Up-/Down Counter UDC 0/1.
P20_0	P20_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN2		Serial data input pin for LIN-USART 2.
	AIN0		8/16-bit up-count input pin of the Up-/Down Counter UDC 0/1.
Port 21			
P21_7	P21_7	TP00_0	General purpose I/O.
P21_6	P21_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK1		Clock I/O pin for LIN-USART 1.
	CK1		Input for the 16-bit I/O Timer FRT 1.
P21_5	P21_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT1		Serial data output pin for LIN-USART 1.
P21_4	P21_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN1		Serial data input pin for LIN-USART 1.
P21_3	P21_3	TP00_0	General purpose I/O.
P21_2	P21_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCK0		Clock I/O pin for LIN-USART 0.
	CK0		Input for the 16-bit I/O Timer FRT 0.
P21_1	P21_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SOT0		Serial data output pin for LIN-USART 0.
P21_0	P21_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SIN0		Serial data input pin for LIN-USART 0.
Port 22			
P22_7	P22_7	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCL1		Serial clock I/O pin for I2C 1.

Pin Name	I/O Signal	Circuit Type	Function
P22_6	P22_6	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SDA1		Serial data I/O pin for I2C 1.
	INT15		External interrupt input pin for INT 15.
P22_5	P22_5	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SCL0		Serial clock I/O pin for I2C 0.
P22_4	P22_4	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SDA0		Serial data I/O pin for I2C 0.
	INT14		External interrupt input pin for INT 14.
P22_3	P22_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX5		Transmission output pin for CAN 5.
P22_2	P22_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX5		Reception input pin for CAN 5.
	INT13		External interrupt input pin for INT 13.
P22_1	P22_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX4		Transmission output pin for CAN 4.
P22_0	P22_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX4		Reception input pin for CAN 4.
	INT12		External interrupt input pin for INT 12.
Port 23			
P23_7	P23_7	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX3		Transmission output pin for CAN 3.
P23_6	P23_6	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX3		Reception input pin for CAN 3.
	INT11		External interrupt input pin for INT 11.
P23_5	P23_5	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX2		Transmission output pin for CAN 2.
P23_4	P23_4	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX2		Reception input pin for CAN 2.
	INT10		External interrupt input pin for INT 10.
P23_3	P23_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX1		Transmission output pin for CAN 1.

Pin Name	I/O Signal	Circuit Type	Function
P23_2	P23_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX1		Reception input pin for CAN 1.
	INT9		External interrupt input pin for INT 9.
P23_1	P23_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	TX0		Transmission output pin for CAN 0.
P23_0	P23_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	RX0		Reception input pin for CAN 0.
	INT8		External interrupt input pin for INT 8.
Port 24			
P24_7	P24_7	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT7		External interrupt input pin for INT 7.
	SCL3		Serial clock I/O pin for I2C 3.
P24_6	P24_6	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT6		External interrupt input pin for INT 6.
	SDA3		Serial data I/O pin for I2C 3.
P24_5	P24_5	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT5		External interrupt input pin for INT 5.
	SCL2		Serial clock I/O pin for I2C 2.
P24_4	P24_4	TP02_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT4		External interrupt input pin for INT 4.
	SDA2		Serial data I/O pin for I2C 2.
P24_3	P24_3	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT3		External interrupt input pin for INT 3.
P24_2	P24_2	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT2		External interrupt input pin for INT 2.
P24_1	P24_1	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT1		External interrupt input pin for INT 1.
P24_0	P24_0	TP00_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	INT0		External interrupt input pin for INT 0.
Port 25			
P25_7	P25_7	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M5		PWM output 2M (-) stepper motor controller 5.

Pin Name	I/O Signal	Circuit Type	Function
P25_6	P25_6	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P5		PWM output 2P (+) stepper motor controller 5.
P25_5	P25_5	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M5		PWM output 1M (-) stepper motor controller 5.
P25_4	P25_4	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P5		PWM output 1P (+) stepper motor controller 5.
P25_3	P25_3	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M4		PWM output 2M (-) stepper motor controller 4.
P25_2	P25_2	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P4		PWM output 2P (+) stepper motor controller 4.
P25_1	P25_1	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M4		PWM output 1M (-) stepper motor controller 4.
P25_0	P25_0	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P4		PWM output 1P (+) stepper motor controller 4.
Port 26			
P26_7	P26_7	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M3		PWM output 2M (-) stepper motor controller 3.
	AN31		Analog input pin 31 for the A/D converter 1.
P26_6	P26_6	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P3		PWM output 2P (+) stepper motor controller 3.
	AN30		Analog input pin 30 for the A/D converter 1.
P26_5	P26_5	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M3		PWM output 1M (-) stepper motor controller 3.
	AN29		Analog input pin 29 for the A/D converter 1.
P26_4	P26_4	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P3		PWM output 1P (+) stepper motor controller 3.
	AN28		Analog input pin 28 for the A/D converter 1.
P26_3	P26_3	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M2		PWM output 2M (-) stepper motor controller 2.
	AN27		Analog input pin 27 for the A/D converter 1.

Pin Name	I/O Signal	Circuit Type	Function
P26_2	P26_2	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P2		PWM output 2P (+) stepper motor controller 2.
	AN26		Analog input pin 26 for the A/D converter 1.
P26_1	P26_1	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M2		PWM output 1M (-) stepper motor controller 2.
	AN25		Analog input pin 25 for the A/D converter 1.
P26_0	P26_0	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P2		PWM output 1P (+) stepper motor controller 2.
	AN24		Analog input pin 24 for the A/D converter 1.
Port 27			
P27_7	P27_7	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M1		PWM output 2M (-) stepper motor controller 1.
	AN23		Analog input pin 23 for the A/D converter 1.
P27_6	P27_6	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P1		PWM output 2P (+) stepper motor controller 1.
	AN22		Analog input pin 22 for the A/D converter 1.
P27_5	P27_5	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M1		PWM output 1M (-) stepper motor controller 1.
	AN21		Analog input pin 21 for the A/D converter 1.
P27_4	P27_4	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P1		PWM output 1P (+) stepper motor controller 1.
	AN20		Analog input pin 20 for the A/D converter 1.
P27_3	P27_3	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2M0		PWM output 2M (-) stepper motor controller 0.
	AN19		Analog input pin 19 for the A/D converter 1.
P27_2	P27_2	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC2P0		PWM output 2P (+) stepper motor controller 0.
	AN18		Analog input pin 18 for the A/D converter 1.
P27_1	P27_1	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1M0		PWM output 1M (-) stepper motor controller 0.
	AN17		Analog input pin 17 for the A/D converter 1.

Pin Name	I/O Signal	Circuit Type	Function
P27_0	P27_0	TP05_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SMC1P0		PWM output 1P (+) stepper motor controller 0.
	AN16		Analog input pin 16 for the A/D converter 1.
Port 28			
P28_7	P28_7	TP01_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN15		Analog input pin 15 for the A/D converter 1.
	DA1		Analog output pin 1 for the D/A converter 1.
P28_6	P28_6	TP01_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN14		Analog input pin 14 for the A/D converter 1.
	DA0		Analog output pin 0 for the D/A converter 1.
P28_5	P28_5	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN13		Analog input pin 13 for the A/D converter 1.
P28_4	P28_4	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN12		Analog input pin 12 for the A/D converter 1.
P28_3	P28_3	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN11		Analog input pin 11 for the A/D converter 1.
P28_2	P28_2	TP03_0	General purpose I/O.This function is enabled by setting the corresponding PFR to '0'.
	AN10		Analog input pin 10 for the A/D converter 1.
P28_1	P28_1	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN9		Analog input pin 9 for the A/D converter 1.
P28_0	P28_0	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN8		Analog input pin 8 for the A/D converter 1.
Port 29			
P29_7	P29_7	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN7		Analog input pin 7 for the A/D converter 1.
P29_6	P29_6	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN6		Analog input pin 6 for the A/D converter 1.
P29_5	P29_5	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN5		Analog input pin 5 for the A/D converter 1.
P29_4	P29_4	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN4		Analog input pin 4 for the A/D converter 1.

Pin Name	I/O Signal	Circuit Type	Function
P29_3	P29_3	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN3		Analog input pin 3 for the A/D converter 1.
P29_2	P29_2	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN2		Analog input pin 2 for the A/D converter 1.
P29_1	P29_1	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN1		Analog input pin 1 for the A/D converter 1.
P29_0	P29_0	TP03_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	AN0		Analog input pin 0 for the A/D converter 1.
Port 30			
P30_7	P30_7	TP08_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	V3		Analog input pin external reference voltage V3 LCD controller.
P30_6	P30_6	TP07_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	V2		Analog input pin external reference voltage V2 LCD controller.
P30_5	P30_5	TP07_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	V1		Analog input pin external reference voltage V1 LCD controller.
P30_4	P30_4	TP07_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	V0		Analog input pin external reference voltage V0 LCD controller.
P30_3	P30_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	COM3		Common driver output pin 3 LCD controller.
P30_2	P30_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	COM2		Common driver output pin 2 LCD controller.
P30_1	P30_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	COM1		Common driver output pin 1 LCD controller.
P30_0	P30_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	COM0		Common driver output pin 0 LCD controller.
Port 31			
P31_7	P31_7	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG39		Segment driver output pin 39 LCD controller.
P31_6	P31_6	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG38		Segment driver output pin 38 LCD controller.

Pin Name	I/O Signal	Circuit Type	Function
P31_5	P31_5	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG37		Segment driver output pin 37 LCD controller.
P31_4	P31_4	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG36		Segment driver output pin 36 LCD controller.
P31_3	P31_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG35		Segment driver output pin 35 LCD controller.
P31_2	P31_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG34		Segment driver output pin 34 LCD controller.
P31_1	P31_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG33		Segment driver output pin 33 LCD controller.
P31_0	P31_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG32		Segment driver output pin 32 LCD controller.
Port 32			
P32_7	P32_7	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG31		Segment driver output pin 31 LCD controller.
P32_6	P32_6	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG30		Segment driver output pin 30 LCD controller.
	SCK15		Clock I/O pin for LIN-USART 15.
P32_5	P32_5	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG29		Segment driver output pin 29 LCD controller.
	SOT15		Serial data output pin for LIN-USART 15.
P32_4	P32_4	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG28		Segment driver output pin 28 LCD controller.
	SIN15		Serial data input pin for LIN-USART 15.
P32_3	P32_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG27		Segment driver output pin 27 LCD controller.
P32_2	P32_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG26		Segment driver output pin 26 LCD controller.
	SCK14		Clock I/O pin for LIN-USART 14.

Pin Name	I/O Signal	Circuit Type	Function
P32_1	P32_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG25		Segment driver output pin 25 LCD controller.
	SOT14		Serial data output pin for LIN-USART 14.
P32_0	P32_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG24		Segment driver output pin 24 LCD controller.
	SIN14		Serial data input pin for LIN-USART 14.
Port 33			
P33_7	P33_7	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG23		Segment driver output pin 23 LCD controller.
P33_6	P33_6	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG22		Segment driver output pin 22 LCD controller.
	SCK13		Clock I/O pin for LIN-USART 13.
P33_5	P33_5	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG21		Segment driver output pin 21 LCD controller.
	SOT13		Serial data output pin for LIN-USART 13.
P33_4	P33_4	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG20		Segment driver output pin 20 LCD controller.
	SIN13		Serial data input pin for LIN-USART 13.
P33_3	P33_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG19		Segment driver output pin 19 LCD controller.
P33_2	P33_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG18		Segment driver output pin 18 LCD controller.
	SCK12		Clock I/O pin for LIN-USART 12.
P33_1	P33_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG17		Segment driver output pin 17 LCD controller.
	SOT12		Serial data output pin for LIN-USART 12.
P33_0	P33_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG16		Segment driver output pin 16 LCD controller.
	SIN12		Serial data input pin for LIN-USART 12.
Port 34			
P34_7	P34_7	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG15		Segment driver output pin 15 LCD controller.

Pin Name	I/O Signal	Circuit Type	Function
P34_6	P34_6	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG14		Segment driver output pin 14 LCD controller.
	SCK11		Clock I/O pin for LIN-USART 11.
P34_5	P34_5	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG13		Segment driver output pin 13 LCD controller.
	SOT11		Serial data output pin for LIN-USART 11.
P34_4	P34_4	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG12		Segment driver output pin 12 LCD controller.
	SIN11		Serial data input pin for LIN-USART 11.
P34_3	P34_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG11		Segment driver output pin 11 LCD controller.
P34_2	P34_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG10		Segment driver output pin 10 LCD controller.
	SCK10		Clock I/O pin for LIN-USART 10.
P34_1	P34_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG9		Segment driver output pin 9 LCD controller.
	SOT10		Serial data output pin for LIN-USART 10.
P34_0	P34_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG8		Segment driver output pin 8 LCD controller.
	SIN10		Serial data input pin for LIN-USART 10.
Port 35			
P35_7	P35_7	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG7		Segment driver output pin 7 LCD controller.
P35_6	P35_6	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG6		Segment driver output pin 6 LCD controller.
	SCK9		Clock I/O pin for LIN-USART 9.
P35_5	P35_5	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG5		Segment driver output pin 5 LCD controller.
	SOT9		Serial data output pin for LIN-USART 9.
P35_4	P35_4	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG4		Segment driver output pin 4 LCD controller.
	SIN9		Serial data input pin for LIN-USART 9.

Pin Name	I/O Signal	Circuit Type	Function
P35_3	P35_3	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG3		Segment driver output pin 3 LCD controller.
P35_2	P35_2	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG2		Segment driver output pin 2 LCD controller.
	SCK8		Clock I/O pin for LIN-USART 8.
P35_1	P35_1	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG1		Segment driver output pin 1 LCD controller.
	SOT8		Serial data output pin for LIN-USART 8.
P35_0	P35_0	TP06_0	General purpose I/O. This function is enabled by setting the corresponding PFR to '0'.
	SEG0		Segment driver output pin 0 LCD controller.
	SIN8		Serial data input pin for LIN-USART 8.

30.2. I/O Port Functions MB91FV460B

MB91FV460B has more ports and a bigger variety of port / resource assignments and port relocation features, depending on the series mode. All the I/Os of MB91FV460B are already explained in [Chapter 3 MB91460 Series Basic Information](#).

For the series modes, please refer to section [3.8.1 MB91460 Series Modes \(Page No.143\)](#).

A listing of the resource function I/O names can be found in section [3.8.2 General Purpose Port and Resource Function IO Names \(Page No.144\)](#).

All I/O port functions of MB91FV460B are explained in section [3.8.3 User Ports in Default Series Mode \(Page No.146\)](#) and following sections.

The I/O circuit types of MB91FV460B can be found in section [3.9. I/O Circuit Types MB91FV460B \(Page No.179\)](#).

MB91460 Series

30.3. I/O Circuit Types

30.3.1 I/O Cell List MB91V460A

Table 30.3-1 I/O cell list MB91V460A

Type	Input			Analog Line	Output Driver	Comment
	Pull Up / Down (50 kOhm)	CMOS (C) CMOS Hysteresis (CH) Automotive (AH)	Input Stop			
TP00_0	Up/Down switch	CH / AH switch	Stop	-	4 mA	-
TP01_0	-	CH / AH switch	Stop	In/Out	4 mA	ADC / DAC
TP02_0	-	CH / AH switch	Stop	-	3 mA	I2C
TP03_0	Up/Down switch	CH / AH switch	Stop	Input	4 mA	AN
TP04_0	Up/Down switch	CH / AH / TTL Switch	Stop	-	4 mA	External Bus
TP05_0	-	CH / AH switch	Stop	Input	30 mA	SMC / ADC
TP06_0	-	CH / AH switch	Stop	-	4 mA	LCD COM / SEG
TP07_0	-	CH / AH switch	Stop	In/Out	4 mA	LCD V0 / V1 / V2
TP08_0	-	CH / AH switch	Stop	In/Out	4 mA	LCD V3
TC00_0	-	CCH	-	-	-	HSTX
TC01_0	Up	CCH	-	-	-	RSTX / NMIX EBREAKX
TC02_0	Up	CCH	-	-	-	INITX
TC02_1	-	CCH	-	-	-	MD
TC10_0	-	-	-	-	8 mA	MONCLK
TO00_0	-	-	Stop	-	-	4MHz Oscillator
TO00_1	-	-	Stop	-	-	4MHz Oscillator
TO01_0	-	-	Stop	-	-	32kHz Oscillator
TO01_1	-	-	Stop	-	-	32kHz Oscillator
TA02_0	-	-	-	Input	-	ALARM

Note: This table shows the I/O cells used for MB91V460. Please refer to the appropriate datasheet for I/O circuits used on flash devices.

Note: The most I/O cells have programmable input levels (CH / AH [/ TTL]). About the programming, see section [“Port Input Level Selection” on page 646](#).

Note: For the programming of input pull-up or pull-down function, see section [“Programmable Pull-Up/Pull Down Resistors” on page 650](#).

30.3.2 I/O Input Voltages (VIL/VIH)

There are various types of input stages. The following table lists the input voltages VIL / VIH.

Type	Description	VIL	VIH
C	CMOS input	0.25 x VDD	0.65 x VDD
CH	CMOS Hysteresis Trigger input (I/O Port)	0.3 x VDD	0.7 x VDD
CCH	CMOS Hysteresis Trigger input (Control)	0.2 x VDD	0.8 x VDD
AH	CMOS Automotive Hysteresis Trigger input	0.5 x VDD	0.8 x VDD
TTL	TTL Input	0.8 V	2.1 V

30.3.3 I/O Cell List MB91FV460B

The I/O circuit types of MB91FV460B can be found in section [3.9. I/O Circuit Types MB91FV460B \(Page No.179\)](#).

There are an overview table in section [3.9.1 I/O Circuit Overview \(Page No.179\)](#) and the set of drawings in [3.9.2 I/O Circuit Description \(Page No.181\)](#).

30.4. Port Register Settings

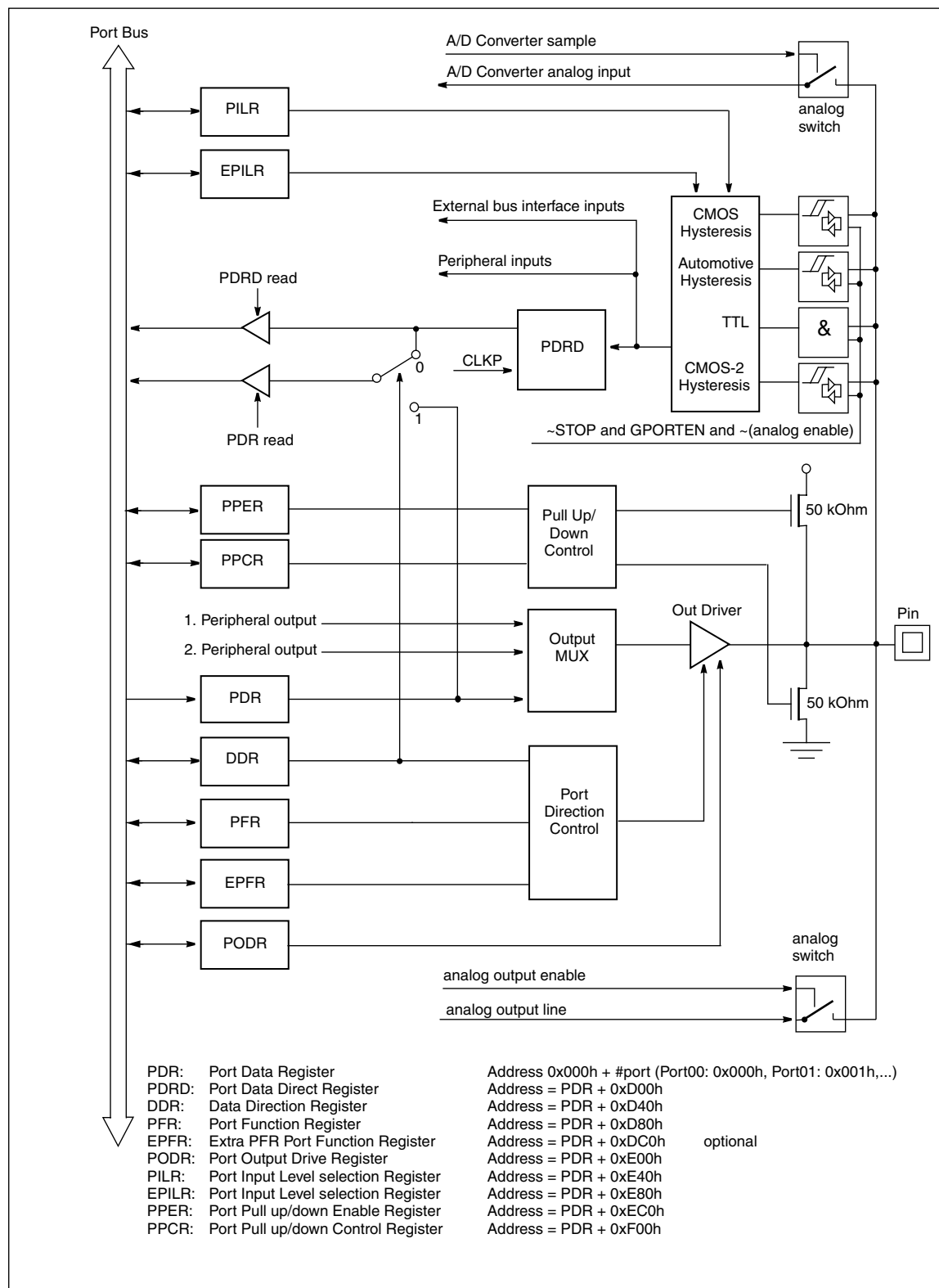
30.4.1 General Rules

For all ports, the following rules are valid:

1. **All port inputs are disabled by default** to avoid transverse current floating before the ports are configured by software. After configuring each port pin according to its function it is necessary to enable the port inputs with the global port enable (PORTEN.GPORTEN). See section [30.4.3 Port Input Enable \(Page No.596\)](#).
2. Each port has a Port Data Register direct read (PDRD) to sample the pin data with CLKP. This register is read-only.
3. Each port has a Data Direction Register (DDR) to switch the port's input/output direction. After reset, all ports are input (DDR=0x00).
 - Port Input mode (PFR = "0" and DDR = "0")
 - PDRD read : Reads the sampled pin data.
 - PDR read : Reads the sampled pin data.
 - PDR write : Writes the PDR setting value, has no effect on the pin value.
 - Port Output mode (PFR = "0" and DDR = "1")
 - PDRD read : Reads the sampled pin data.
 - PDR read : Reads the PDR register value.
 - PDR write : Writes the PDR setting value to the corresponding external pins.
4. On a **Read-Modify-Write** instruction (bit operations) always the PDR register is read independent of the Data Direction Register (DDR) settings.
5. Each port has 2 Port Input Level Registers (PILR and EPILR) to bit-wise select the input level (CMOS-Hysteresis / Automotive / TTL / CMOS-Hysteresis-2). The default value depends on the function of the port. The input level can be set in every device mode. See section [30.4.5 Port Input Level Selection \(Page No.646\)](#).
6. All ports have programmable Pull-Ups/Pull-Downs (50 kOhm) which are enabled bit-wise by their Pull-Up/Pull-Down Enable Registers (PPER) and Pull-Up/Pull-Down Control Registers (PPCR). See section [30.4.6 Programmable Pull-Up/Pull Down Resistors \(Page No.650\)](#).
7. Each port has one or two Port Function Registers: PFR and, if necessary, Extra PFR (EPFR). Together, they can serve up to 3 resource I/O's per pin. See section [30.4.4 Port Function Register Setup \(Page No.598\)](#).
8. Port setup controlled by the MD[2:0] pins and the mode register MODR overwrites the setup in the port registers. E.g. External Bus Mode overwrites port register setup. The external bus signal output can be disabled by setting the PFR of the pin to port mode (PFR='0').
9. **Resource input** lines are generally connected to the pin and are enabled by setting the appropriate functionality inside the resource. There are exceptions which are listed in [30.4.4 Port Function Register Setup \(Page No.598\)](#).
10. **External Interrupt input** lines are always connected to the pin and are enabled in the External Interrupt unit. Care must be taken regarding enabling of external interrupts in STOP mode.
11. **In STOP state** (STCR:STOP set and STCR:HIZ not set) all pins keep their state (input or output depending on the configuration before entering the STOP state) and the input stages and lines are internally fixed to avoid transverse current. External interrupt input pins are not fixed if the corresponding pin is selected by using the PFR='1' setting and the corresponding external interrupt request is enabled with the ENIR0, resp. ENIR1 registers. Pull-Ups and Pull-Downs are enabled.
12. **In STOP-HIZ state** (STCR:STOP and STCR:HIZ set) all pins are switching to input (high impedance state) and all input stages and lines are internally fixed to avoid floating. External interrupt input pins are not fixed if the corresponding pin is selected by using the PFR='1' setting and the corresponding external interrupt request is enabled with the ENIR0, resp. ENIR1 registers. Pull-Ups and Pull-Downs are disabled.
13. **Resource output** lines are enabled by setting the corresponding PFR and/or EPFR bit in the port. Details

see section [30.4.4 Port Function Register Setup \(Page No.598\)](#). LIN-USART outputs (SOT) must be enabled additionally by setting the SOE bit in the LIN-USART control.

14. **Resource bidirectional** signals (e.g. SCK of the LIN-USART) are enabled by setting the corresponding PFR and/or EPFR bit in the port. The signal direction is controlled by the setup of the resource, e.g. via the output enable bits. Details see section [30.4.4 Port Function Register Setup \(Page No.598\)](#).
15. If **analog output**, for example D/A converter or LCD output, is enabled, the digital output driver is set to HIZ by hardware and the digital input lines are disabled by hardware.
16. If **analog input**, for example A/D converter input, is enabled, the digital input lines are disabled by hardware. On old devices, the digital output lines are disabled by hardware too. On new devices (MB91FV460B or later), the digital output lines can be kept active to measure external shortages. See [30.4.3 Port Input Enable \(Page No.596\)](#).



30.4.3 Port Input Enable

This section describes the Port Input Enable function.

■ PORTEN: Port Input Enable.

Addr	7	6	5	4	3	2	1	0	initial
PORTEN 0498h	-	-	-	-	-	ADCHE	CPORTEN	GPORTEN	---- -000
	-	-	-	-	-	R/W	R/W	R/W	

All port inputs are disabled by default to avoid transverse current floating in the IO input stages and the subsequent logic. After configuring all ports according to their functional specification (input level, output drive, pull-up or pull-down resistor, etc.) it is mandatory to globally enable the inputs by setting the port input enable bit.

GPORTEN 0 - The inputs of all ports are disabled.
1 - The inputs of all ports are enabled.

CPORTEN 0 - The inputs of the bootloader communication ports are disabled.
1 - The inputs of the bootloader communication ports are enabled.

Analog channels disable digital I/O, if they are enabled. On new devices (MB91FV460B or later), the digital output lines can be kept active to measure external shortages. The bit ADCHE, together with the ADC channel enable bits ADEn, controls this behaviour.

ADCHE 0 - Global A/D Channel Enable is OFF.
1 - Global A/D Channel Enable is ON.

Table 30.4-1 Function of PORTEN.ADCHE bit

ADERH.ADEn ADERL.ADEn ¹	PORTEN. ADCHE	Function
0 [initial]	X	Analog input of A/D channel n is disabled. The ADC will not sample/convert this channel and the digital I/O operation is enabled.
1	0 [initial]	Analog input of the channel n is enabled. Additionally, the port function register (PFR,EPFR) of the corresponding port must be set . The PFR/EPFR will switch the port to input direction (output driver = HiZ) and disable the digital input lines.
	1	Analog input of the channel n is enabled. Setting the port function register(s) is not necessary. ADEn will disable the digital input lines of the ports, but it does not change the ports direction.

1. n = 0 to 31 depending on available ADC channels on the device.

Note: For new ADC channels (AN32 to AN53, device depending), the ADCHE feature is always ON.

For old ADC channels (AN0 to AN31), the ADCHE feature is always OFF if the channels are re-located to other pins (in Series Modes B,P/T). On old devices (before MB91FV460B) the ADCHE feature is not implemented.

■ General I/O of the external bus interface:

On new devices, the external bus interface is disabled after reset in internal vector fetch mode. See section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#). In this case, GPORTEN is valid to enable the input lines.

On devices with external bus interface enabled after reset, the following is valid:

MB91460 Series

- **Output lines:**

The external bus interface is initialized by default after power-on.

Hence, the address lines and the relevant control lines are set to output and high after power-on.

- **Input lines:**

The input lines of the external bus interface are enabled independent from GPORTEN by the following constellations:

External Vector Fetch (MD2-0=001):

At Vector Fetch: D31-24 are enabled

After Vector Fetch:

Value of Mode-Vector (WTH1-0)	Enabled Data Lines of the external bus
00 (byte)	D31-24
01 (halfword)	D31-16
10 (word)	D31-00
11 (single)	none

Internal Vector Fetch (MD2-0=000):

After Vector Fetch:

D31-00 are enabled, because of Fixed-Mode-Vector-WTH1-0 = 10 (word).

30.4.4 Port Function Register Setup

This section describes the Port Function Registers of each port.

■ **P00:** The functions of Port 00 are controlled by PFR00

	Addr	7	6	5	4	3	2	1	0	initial
PFR00	0D80h	PFR00.7	PFR00.6	PFR00.5	PFR00.4	PFR00.3	PFR00.2	PFR00.1	PFR00.0	#### #### ¹
EPFR00	0DC0h	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P00[7:0]** is input/output for data lines **D[31:24]**. Otherwise, the port can be used as general purpose port.

PFR00.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR00.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

MB91460 Series**■ P01:** The functions of Port 01 are controlled by PFR01

	Addr	7	6	5	4	3	2	1	0	initial
PFR01	0D81h	PFR01.7	PFR01.6	PFR01.5	PFR01.4	PFR01.3	PFR01.2	PFR01.1	PFR01.0	#### #### ¹
EPFR01	0DC1h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P01[7:0]** is input/output for data lines **D[23:16]**. Otherwise, the port can be used as general purpose port.

PFR01.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR01.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

■ **P02:** The functions of Port 02 are controlled by PFR02

	Addr	7	6	5	4	3	2	1	0	initial
PFR02	0D82h	PFR02.7	PFR02.6	PFR02.5	PFR02.4	PFR02.3	PFR02.2	PFR02.1	PFR02.0	#### #### ¹
EPFR02	0DC2h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P02[7:0]** is input/output for data lines **D[15:8]**. Otherwise, the port can be used as general purpose port.

PFR02.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR02.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

MB91460 Series**■ P03:** The functions of Port 03 are controlled by PFR03

	Addr	7	6	5	4	3	2	1	0	initial
PFR03	0D83h	PFR03.7	PFR03.6	PFR03.5	PFR03.4	PFR03.3	PFR03.2	PFR03.1	PFR03.0	#### #### ¹
EPFR03	0DC3h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P03[7:0]** is input/output for data lines **D[7:0]**. Otherwise, the port can be used as general purpose port.

PFR03.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR03.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

■ **P04:** The functions of Port 04 are controlled by PFR04

	Addr	7	6	5	4	3	2	1	0	initial
PFR04	0D84h	PFR04.7	PFR04.6	PFR04.5	PFR04.4	PFR04.3	PFR04.2	PFR04.1	PFR04.0	#### #### ¹
EPFR04	0DC4h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P04[7:0]** is input/output for address lines **A[31:24]**. Otherwise, the port can be used as general purpose port.

PFR04.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR04.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

MB91460 Series**■ P05:** The functions of Port 05 are controlled by PFR05

	Addr	7	6	5	4	3	2	1	0	initial
PFR05	0D85h	PFR05.7	PFR05.6	PFR05.5	PFR05.4	PFR05.3	PFR05.2	PFR05.1	PFR05.0	#### #### ¹
EPFR05	0DC5h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P05[7:0]** is input/output for address lines **A[23:16]**. Otherwise, the port can be used as general purpose port.

PFR05.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR05.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

■ **P06:** The functions of Port 06 are controlled by PFR06

	Addr	7	6	5	4	3	2	1	0	initial
PFR06	0D86h	PFR06.7	PFR06.6	PFR06.5	PFR06.4	PFR06.3	PFR06.2	PFR06.1	PFR06.0	#### #### ¹
EPFR06	0DC6h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P06[7:0]** is input/output for address lines **A[15:8]**. Otherwise, the port can be used as general purpose port.

PFR06.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR06.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

MB91460 Series**■ P07:** The functions of Port 07 are controlled by PFR07

	Addr	7	6	5	4	3	2	1	0	initial
PFR07	0D87h	PFR07.7	PFR07.6	PFR07.5	PFR07.4	PFR07.3	PFR07.2	PFR07.1	PFR07.0	#### #### ¹
EPFR07	0DC7h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P07[7:0]** is input/output for address lines **A[7:0]**. Otherwise, the port can be used as general purpose port.

PFR07.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR07.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

■ **P08:** The functions of Port 08 are controlled by PFR08

	Addr	7	6	5	4	3	2	1	0	initial
PFR08	0D88h	PFR08.7	PFR08.6	PFR08.5	PFR08.4	PFR08.3	PFR08.2	PFR08.1	PFR08.0	#### #### ¹
EPFR08	0DC8h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P08[7:0]** is input/output for external bus control signals **RDY**, **BRQ**, **BGRNTX**, **RDX**, **WRX[3:0]**. Otherwise, the port can be used as general purpose port.

- PFR08.7** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **RDY**
- PFR08.6** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **BRQ**
- PFR08.5** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **BGRNTX**
- PFR08.4** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **RDX**
- PFR08.3** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **WRX3**
- PFR08.2** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **WRX2**
- PFR08.1** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **WRX1**
- PFR08.0** 0 - Port is in general purpose port mode.
 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):
 External bus function is **WRX0**

MB91460 Series**■ P09:** The functions of Port 09 are controlled by PFR09

	Addr	7	6	5	4	3	2	1	0	initial
PFR09	0D89h	PFR09.7	PFR09.6	PFR09.5	PFR09.4	PFR09.3	PFR09.2	PFR09.1	PFR09.0	#### #### ¹
EPFR09	0DC9h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P09[7:0]** is input/output for external bus control signals **CSX[7:0]**. Otherwise, the port can be used as general purpose port.

PFR09.7	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.6	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.5	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.4	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.3	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.2	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.1	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).
PFR09.0	0 - Port is in general purpose port mode. 1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port).

■ **P10:** The functions of Port 10 are controlled by PFR10 and EPFR10

	Addr	7	6	5	4	3	2	1	0	initial
PFR10	0D8Ah	-	PFR10.6	PFR10.5	PFR10.4	PFR10.3	PFR10.2	PFR10.1	PFR10.0	-#0# #### ¹
EPFR10	0DCAh	-	-	EPFR10.5	EPFR10.4	-	-	-	EPFR10.0	--00 ---0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

1. Device dependent, see section [3.10.3 Initial Behaviour of the External Bus \(Page No.193\)](#).

If the external bus interface is enabled (by mode pins MD[2:0] or mode vector), **P10[7:0]** is input/output for external bus control signals **MCLKE**, **MCLKI**, **MCLKO**, **WEX**, **BAAX**, **ASX**, **SYSCLK**. Otherwise, the port can be used as general purpose port.

PFR10.6 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

External bus function is **MCLKE**

PFR10.5 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

EPFR10.5 0 - External bus function is **MCLKI**
1 - External bus function is **MCLKIX** (inverted input)

PFR10.4 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

EPFR10.4 0 - External bus function is **MCLKO**
1 - External bus function is **MCLKOX** (inverted output)

PFR10.3 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

External bus function is **WEX**

PFR10.2 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

External bus function is **BAAX**

PFR10.1 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

External bus function is **ASX**

PFR10.0 0 - Port is in general purpose port mode.
1 - Port is in external bus mode (if external bus is enabled otherwise general purpose port):

EPFR10.4 0 - External bus function is **SYSCLK**
1 - External bus function is **SYSCLKX** (inverted output)

MB91460 Series**■ P11:** The functions of Port 11 are controlled by PFR11

	Addr	7	6	5	4	3	2	1	0	initial
PFR11	0D8Bh	-	-	-	-	-	-	PFR11.1	PFR11.0	---- --00
EPFR11	0DCBh	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P11[7:0] is input/output for DMA control signals **IOWRX**, **IORDX**. Otherwise, the port can be used as general purpose port.

PFR11.1 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **IOWRX** output

PFR11.0 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **IORDX** output

■ **P12:** The functions of Port 12 are controlled by PFR12 and EPFR12

	Addr	7	6	5	4	3	2	1	0	initial
PFR12	0D8Ch	PFR12.7	PFR12.6	PFR12.5	PFR12.4	PFR12.3	PFR12.2	PFR12.1	PFR12.0	0000 0000
EPFR12	0DCCh	-	EPFR12.6	-	-	-	EPFR12.2	-	-	-0-- -0--
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P12[7:0] is input/output for DMA control signals **DEOP**, **DEOTX**, **DACKX**, **DREQ** for DMA channels 2 and 3. Otherwise, the port can be used as general purpose port.

PFR12.7	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DEOP3 output
PFR12.6	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: EPFR12.6 0 - DMA function is DEOTX3 input 1 - DMA function is DEOP3 output
PFR12.5	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DACKX3 output
PFR12.4	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DREQ3 input
PFR12.3	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DEOP2 output
PFR12.2	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: EPFR12.2 0 - DMA function is DEOTX2 input 1 - DMA function is DEOP2 output
PFR12.1	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DACKX2 output
PFR12.0	0 - Port is in general purpose port mode. 1 - Port is in DMA function mode: DMA function is DREQ2 input

MB91460 Series

■ **P13:** The functions of Port 13 are controlled by PFR13 and EPFR13

	Addr	7	6	5	4	3	2	1	0	initial
PFR13	0D8Dh	PFR13.7	PFR13.6	PFR13.5	PFR13.4	PFR13.3	PFR13.2	PFR13.1	PFR13.0	0000 0000
EPFR13	0DCDh	-	EPFR13.6	-	-	-	EPFR13.2	-	-	-0-- -0--
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P13[7:0] is input/output for DMA control signals **DEOP**, **DEOTX**, **DACKX**, **DREQ** for DMA channels 0 and 1. Otherwise, the port can be used as general purpose port.

PFR13.7 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DEOP1** output

PFR13.6 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 EPFR13.6 0 - DMA function is **DEOTX1** input
 1 - DMA function is **DEOP1** output

PFR13.5 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DACKX1** output

PFR13.4 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DREQ1** input

PFR13.3 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DEOP0** output

PFR13.2 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 EPFR13.2 0 - DMA function is **DEOTX0** input
 1 - DMA function is **DEOP0** output

PFR13.1 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DACKX0** output

PFR13.0 0 - Port is in general purpose port mode.
 1 - Port is in DMA function mode:

 DMA function is **DREQ0** input

■ **P14:** The functions of Port 14 are controlled by PFR14 and EPFR14

	Addr	7	6	5	4	3	2	1	0	initial
PFR14	0D8Eh	PFR14.7	PFR14.6	PFR14.5	PFR14.4	PFR14.3	PFR14.2	PFR14.1	PFR14.0	0000 0000
EPFR14	0DCEh	EPFR14.7	EPFR14.6	EPFR14.5	EPFR14.4	EPFR14.3	EPFR14.2	EPFR14.1	EPFR14.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P14[7:0] is input for Input Capture inputs **ICU[7:0]**, Reload Timer triggers **TIN[15:0]** and PWM inputs **TTG[31:0]**. Otherwise, the port can be used as general purpose port and the ICU modules are connected to the LSYN outputs of 2 LIN-UARTs (whereas the 2 LSYN lines are connected by OR function).

Additionally, P14[7] is input for the FlexRay Stop Watch signal **STOPWT**.

- PFR14.7** 0 - Port is in general purpose port mode. ICU7 is internally connected to LSYN of LIN-UART 7/15.
 1 - Port is in resource function mode:
 Resource function is **STOPWT**, **TIN15/7** and **TTG31/23/15/7** input, and
 EPFR14.7 0 - Resource function is **ICU7** input
 1 - ICU7 is internally connected to LSYN of LIN-UART 7/15
- PFR14.6** 0 - Port is in general purpose port mode. ICU6 is internally connected to LSYN of LIN-UART 6/14.
 1 - Port is in resource function mode:
 Resource function is **TIN14/6** and **TTG30/22/14/6** input, and
 EPFR14.6 0 - Resource function is **ICU6** input
 1 - ICU6 is internally connected to LSYN of LIN-UART 6/14
- PFR14.5** 0 - Port is in general purpose port mode. ICU5 is internally connected to LSYN of LIN-UART 5/13.
 1 - Port is in resource function mode:
 Resource function is **TIN13/5** and **TTG29/21/13/5** input, and
 EPFR14.5 0 - Resource function is **ICU5** input
 1 - ICU5 is internally connected to LSYN of LIN-UART 5/13
- PFR14.4** 0 - Port is in general purpose port mode. ICU4 is internally connected to LSYN of LIN-UART 4/12.
 1 - Port is in resource function mode:
 Resource function is **TIN12/4** and **TTG28/20/12/4** input, and
 EPFR14.4 0 - Resource function is **ICU4** input
 1 - ICU4 is internally connected to LSYN of LIN-UART 4/12
- PFR14.3** 0 - Port is in general purpose port mode. ICU3 is internally connected to LSYN of LIN-UART 3/11.
 1 - Port is in resource function mode:
 Resource function is **TIN11/3** and **TTG27/19/11/3** input, and
 EPFR14.3 0 - Resource function is **ICU3** input
 1 - ICU3 is internally connected to LSYN of LIN-UART 3/11
- PFR14.2** 0 - Port is in general purpose port mode. ICU2 is internally connected to LSYN of LIN-UART 2/10.
 1 - Port is in resource function mode:
 Resource function is **TIN10/2** and **TTG26/18/10/2** input, and
 EPFR14.2 0 - Resource function is **ICU2** input
 1 - ICU2 is internally connected to LSYN of LIN-UART 2/10

MB91460 Series

PFR14.1 0 - Port is in general purpose port mode. ICU1 is internally connected to LSYN of LIN-UART 1/9.
 1 - Port is in resource function mode:

Resource function is **TIN9/1** and **TTG25/17/9/1** input, and

EPFR14.1 0 - Resource function is **ICU1** input
 1 - ICU1 is internally connected to LSYN of LIN-UART 1/9

PFR14.0 0 - Port is in general purpose port mode. ICU0 is internally connected to LSYN of LIN-UART 0/8.
 1 - Port is in resource function mode:

Resource function is **TIN8/0** and **TTG24/16/8/0** input, and

EPFR14.0 0 - Resource function is **ICU0** input
 1 - ICU0 is internally connected to LSYN of LIN-UART 0/8

Remark: It is generally possible to use the input only resource functions TIN, TTG and STOPWT also in the general purpose port input mode (PFR='0' and DDR='0').

■ **P15:** The functions of Port 15 are controlled by PFR15 and EPFR15

	Addr	7	6	5	4	3	2	1	0	initial
PFR15	0D8Fh	PFR15.7	PFR15.6	PFR15.5	PFR15.4	PFR15.3	PFR15.2	PFR15.1	PFR15.0	0000 0000
EPFR15	0DCFh	EPFR15.7	EPFR15.6	EPFR15.5	EPFR15.4	EPFR15.3	EPFR15.2	EPFR15.1	EPFR15.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P15[7:0] is output for Output Compare outputs **OCU[7:0]** and Reload Timer outputs **TOT[7:0]**. Otherwise, the port can be used as general purpose port.

PFR15.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.7 0 - Resource function is OCU7 output 1 - Resource function is TOT7 output
PFR15.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.6 0 - Resource function is OCU6 output 1 - Resource function is TOT6 output
PFR15.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.5 0 - Resource function is OCU5 output 1 - Resource function is TOT5 output
PFR15.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.4 0 - Resource function is OCU4 output 1 - Resource function is TOT4 output
PFR15.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.3 0 - Resource function is OCU3 output 1 - Resource function is TOT3 output
PFR15.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.2 0 - Resource function is OCU2 output 1 - Resource function is TOT2 output
PFR15.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.1 0 - Resource function is OCU1 output 1 - Resource function is TOT1 output
PFR15.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR15.0 0 - Resource function is OCU0 output 1 - Resource function is TOT0 output

MB91460 Series

■ **P16:** The functions of Port 16 are controlled by PFR16 and EPFR16

	Addr	7	6	5	4	3	2	1	0	initial
PFR16	0D90h	PFR16.7	PFR16.6	PFR16.5	PFR16.4	PFR16.3	PFR16.2	PFR16.1	PFR16.0	0000 0000
EPFR16	0DD0h	EPFR16.7	EPFR16.6	EPFR16.5	EPFR16.4	-	-	-	-	0000 ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P16[7:0] is input/output for Programmable Pulse Generator outputs **PPG[15:8]**, external ADC trigger **ATGX0/1**, Pulse Frequency Modulator output **PFM**, and Sound Generator outputs **SGO/SGA**. Otherwise, the port can be used as general purpose port.

PFR16.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR16.7 0 - Resource function is PPG15 output 1 - Resource function is ATGX0/1 input (for both ADC0 and ADC1)
PFR16.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR16.6 0 - Resource function is PPG14 output 1 - Resource function is PFM output
PFR16.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR16.5 0 - Resource function is PPG13 output 1 - Resource function is SGO output
PFR16.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR16.4 0 - Resource function is PPG12 output 1 - Resource function is SGA output
PFR16.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is PPG11 output
PFR16.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is PPG10 output
PFR16.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is PPG9 output
PFR16.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is PPG8 output

Remark: It is generally possible to use the input only resource functions ATGX0/1 also in the general purpose port input mode (PFR='0' and DDR='0').

■ **P17:** The functions of Port 17 are controlled by PFR17 and EPFR17

	Addr	7	6	5	4	3	2	1	0	initial
PFR17	0D91h	PFR17.7	PFR17.6	PFR17.5	PFR17.4	PFR17.3	PFR17.2	PFR17.1	PFR17.0	0000 0000
EPFR17	0DD1h	EPFR17.7	EPFR17.6	EPFR17.5	EPFR17.4	-	EPFR17.2	EPFR17.1	EPFR17.0	0000 -000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P17[7:0] is input/output for Programmable Pulse Generator outputs **PPG[15:8]** or for **APIX[®]** sideband, and A/D converter analogue inputs **AN[39:32]**. Otherwise, the port can be used as general purpose port.

The A/D converter analogue inputs AN[39:32] are enabled independently of EPFR/PFR by setting the corresponding ADC channel enable bits in AD1ERL[7:0]. If the ADC channel is enabled, the digital input lines to port and resources are disabled, while the port output lines are maintained if the port was in output direction.

PFR17.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.7 0 - Resource function is PPG7 output 1 - Resource function is APIX TEN1 input
PFR17.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.6 0 - Resource function is PPG6 output 1 - Resource function is APIX TDA11 output
PFR17.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.5 0 - Resource function is PPG5 output 1 - Resource function is APIX TDA10 output
PFR17.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.4 0 - Resource function is PPG4 output 1 - Resource function is APIX TCK1 output
PFR17.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is PPG3 output
PFR17.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.2 0 - Resource function is PPG2 output 1 - Resource function is APIX RDA11 input
PFR17.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.1 0 - Resource function is PPG1 output 1 - Resource function is APIX RDA10 input
PFR17.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR17.0 0 - Resource function is PPG0 output 1 - Resource function is APIX RCK1 input

MB91460 Series

■ **P18:** The functions of Port 18 are controlled by PFR18 and EPFR18

	Addr	7	6	5	4	3	2	1	0	initial
PFR18	0D92h	-	PFR18.6	PFR18.5	PFR18.4	-	PFR18.2	PFR18.1	PFR18.0	-000 -000
EPFR18	0DD2h	-	EPFR18.6	EPFR18.5	EPFR18.4	-	EPFR18.2	EPFR18.1	EPFR18.0	-000 -000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P18[7:0] is input/output for LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 6 and 7, Up-/Down-Counter inputs **ZIN**, **BIN**, **AIN** of channels 2 and 3, and Free-Run Timer FRT inputs **CK** of channels 6 and 7, and A/D converter analogue inputs **AN[47:40]**. Otherwise, the port can be used as general purpose port.

The A/D converter analogue inputs AN[47:40] are enabled independently of EPFR/PFR by setting the corresponding ADC channel enable bits in AD1ERL[15:8]. If the ADC channel is enabled, the digital input lines to port and resources are disabled, while the port output lines are maintained if the port was in output direction.

PFR18.6 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.6 0 - Resource function is **SCK7** input/output

1 - Resource function is **ZIN3** and **CK7** input

PFR18.5 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.5 0 - Resource function is **SOT7** output

1 - Resource function is **BIN3** input

PFR18.4 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.4 has no effect. Resource function is **SIN7** and **AIN3** input

PFR18.2 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.2 0 - Resource function is **SCK6** input/output

1 - Resource function is **ZIN2** and **CK6** input

PFR18.1 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.1 0 - Resource function is **SOT6** output

1 - Resource function is **BIN2** input

PFR18.0 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR18.0 has no effect. Resource function is **SIN6** and **AIN2** input

Remark: It is generally possible to use the input only resource functions AIN, BIN, ZIN, CK, SIN also in the general purpose port input mode (PFR='0' and DDR='0' and ADC channel disabled).

■ **P19:** The functions of Port 19 are controlled by PFR19 and EPFR19

	Addr	7	6	5	4	3	2	1	0	initial
PFR19	0D93h	-	PFR19.6	PFR19.5	PFR19.4	-	PFR19.2	PFR19.1	PFR19.0	-000 -000
EPFR19	0DD3h	-	EPFR19.6	-	-	-	EPFR19.2	-	-	-0-- -0--
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P19[7:0] is input/output for LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 4 and 5, and Free-Run Timer FRT inputs **CK** of channels 4 and 5. Otherwise, the port can be used as general purpose port.

P19[7] and **P19[3]** can be used as A/D converter analogue inputs **AN49** and **AN48**. The A/D converter analogue inputs are enabled independently of EPFR/PFR by setting the corresponding ADC channel enable bits in AD1ERH[1:0]. If the ADC channel is enabled, the digital input lines to port and resources are disabled, while the port output lines are maintained if the port was in output direction.

PFR19.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR19.6 0 - Resource function is SCK5 input/output 1 - Resource function is CK5 input
PFR19.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SOT5 output
PFR19.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SIN5 input
PFR19.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR19.2 0 - Resource function is SCK4 input/output 1 - Resource function is CK4 input
PFR19.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SOT4 output
PFR19.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SIN4 input

Remark: It is generally possible to use the input only resource functions CK and SIN also in the general purpose port input mode (PFR='0' and DDR='0').

MB91460 Series

■ **P20:** The functions of Port 20 are controlled by PFR20 and EPFR20

	Addr	7	6	5	4	3	2	1	0	initial
PFR20	0D94h	-	PFR20.6	PFR20.5	PFR20.4	-	PFR20.2	PFR20.1	PFR20.0	-000 -000
EPFR20	0DD4h	-	EPFR20.6	EPFR20.5	EPFR20.4	-	EPFR20.2	EPFR20.1	EPFR20.0	-000 -000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P20[7:0] is input/output for LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 2 and 3, Up-/Down-Counter inputs **ZIN**, **BIN**, **AIN** of channels 0 and 1, and Free-Run Timer FRT inputs **CK** of channels 2 and 3. Otherwise, the port can be used as general purpose port.

P20[7] and **P20[3]** can be used as A/D converter analogue inputs **AN51** and **AN50**. The A/D converter analogue inputs are enabled independently of EPFR/PFR by setting the corresponding ADC channel enable bits in AD1ERH[3:2]. If the ADC channel is enabled, the digital input lines to port and resources are disabled, while the port output lines are maintained if the port was in output direction.

PFR20.6 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.6 0 - Resource function is **SCK3** input/output

1 - Resource function is **ZIN1** and **CK3** input

PFR20.5 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.5 0 - Resource function is **SOT3** output

1 - Resource function is **BIN1** input

PFR20.4 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.4 has no effect. Resource function is **SIN3** and **AIN1** input

PFR20.2 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.2 0 - Resource function is **SCK2** input/output

1 - Resource function is **ZIN0** and **CK2** input

PFR20.1 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.1 0 - Resource function is **SOT2** output

1 - Resource function is **BIN0** input

PFR20.0 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

EPFR20.0 Resource function is **SIN2** and **AIN0** input

Remark: It is generally possible to use the input only resource functions AIN, BIN, ZIN, CK, SIN also in the general purpose port input mode (PFR='0' and DDR='0').

■ **P21:** The functions of Port 21 are controlled by PFR21 and EPFR21

	Addr	7	6	5	4	3	2	1	0	initial
PFR21	0D95h	-	PFR21.6	PFR21.5	PFR21.4	-	PFR21.2	PFR21.1	PFR21.0	-000 -000
EPFR21	0DD5h	-	EPFR21.6	-	-	-	EPFR21.2	-	-	-0-- -0--
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P21[7:0] is input/output for LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 0 and 1, and Free-Run Timer FRT inputs **CK** of channels 0 and 1. Otherwise, the port can be used as general purpose port.

P21[7] and **P21[3]** can be used as A/D converter analogue inputs **AN53** and **AN52**. The A/D converter analogue inputs are enabled independently of EPFR/PFR by setting the corresponding ADC channel enable bits in AD1ERH[5:4]. If the ADC channel is enabled, the digital input lines to port and resources are disabled, while the port output lines are maintained if the port was in output direction.

PFR21.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR21.6 0 - Resource function is SCK1 input/output 1 - Resource function is CK1 input
PFR21.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SOT1 output
PFR21.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SIN1 input
PFR21.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR21.2 0 - Resource function is SCK0 input/output 1 - Resource function is CK0 input
PFR21.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SOT0 output
PFR21.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SIN0 input

Remark: It is generally possible to use the input only resource functions CK and SIN also in the general purpose port input mode (PFR='0' and DDR='0').

MB91460 Series**■ P22:** The functions of Port 22 are controlled by PFR22

	Addr	7	6	5	4	3	2	1	0	initial
PFR22	0D96h	PFR22.7	PFR22.6	PFR22.5	PFR22.4	PFR22.3	PFR22.2	PFR22.1	PFR22.0	0000 0000
EPFR22	0DD6h	-	-	-	-	EPFR22.3	-	EPFR22.1	-	---- 0-0-
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P22[7:0] is input/output for I2C serial communication signals **SCL**, **SDA** of channels 0 and 1, CAN serial communication signals **TX**, **RX** of channels 4 and 5, and External Interrupt Triggers **INT[15:12]**. Otherwise, the port can be used as general purpose port.

P22[7:6] can be used as input lines for **ICU8** and **ICU9**.

PFR22.7 0 - Port is in general purpose port mode. **ICU9** input is connected to the port input line.
1 - Port is in resource function mode:

Resource function is **SCL1** open drain.

ICU9 is internally connected to the LSYN output of LIN-UART9.

PFR22.6 0 - Port is in general purpose port mode. **ICU8** input is connected to the port input line.
1 - Port is in resource function mode:

Resource function is **SDA1** open drain, and **INT15** input

ICU8 is internally connected to the LSYN output of LIN-UART8.

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN7 set to '1'.

PFR22.5 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SCL0** open drain

PFR22.4 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SDA0** open drain, and **INT14** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN6 set to '1'.

PFR22.3 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **TX5** output

PFR22.2 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **RX5** input, and **INT13** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN5 set to '1'.

PFR22.1 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **TX4** output

PFR22.0 0 - Port is in general purpose port mode.

1 - Port is in resource function mode:

Resource function is **RX4** input, and **INT12** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN4 set to '1'.

Remark: It is generally possible to use input only resource functions INT, RX also in the general purpose port input mode (PFR='0' and DDR='0'). In that case, wakeup from STOP state by external interrupt is not possible.

MB91460 Series**■ P23:** The functions of Port 23 are controlled by PFR23

	Addr	7	6	5	4	3	2	1	0	initial
PFR23	0D97h	PFR23.7	PFR23.6	PFR23.5	PFR23.4	PFR23.3	PFR23.2	PFR23.1	PFR23.0	0000 0000
EPFR23	0DD7h	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P23[7:0] is input/output for CAN serial communication signals **TX**, **RX** of channels 0 to 3, and External Interrupt Triggers **INT[11:8]**. Otherwise, the port can be used as general purpose port.

PFR23.7 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **TX3** output

PFR23.6 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **RX3** input, and **INT11** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN3 set to '1'.

PFR23.5 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **TX2** output

PFR23.4 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **RX2** input, and **INT10** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN2 set to '1'.

PFR23.3 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **TX1** output

PFR23.2 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **RX1** input, and **INT9** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN1 set to '1'.

PFR23.1 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **TX0** output

PFR23.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **RX0** input, and **INT8** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR1.EN0 set to '1'.

Remark: It is generally possible to use input only resource functions INT, RX also in the general purpose port input mode (PFR='0' and DDR='0'). In that case, wakeup from STOP state by external interrupt is not possible.

MB91460 Series**■ P24:** The functions of Port 24 are controlled by PFR24

	Addr	7	6	5	4	3	2	1	0	initial
PFR24	0D98h	PFR24.7	PFR24.6	PFR24.5	PFR24.4	PFR24.3	PFR24.2	PFR24.1	PFR24.0	0000 0000
EPFR24	0DD8h	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P24[7:0] is input/output for I2C serial communication signals **SCL**, **SDA** of channels 2 and 3, and External Interrupt Triggers **INT[7:0]**. Otherwise, the port can be used as general purpose port.

PFR24.7 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SCL3** open drain, and **INT7** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN7 set to '1'.

PFR24.6 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SDA3** open drain, and **INT6** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN6 set to '1'.

PFR24.5 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SCL2** open drain, and **INT5** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN5 set to '1'.

PFR24.4 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **SDA2** open drain, and **INT4** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN4 set to '1'.

PFR24.3 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **INT3** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN3 set to '1'.

PFR24.2 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **INT2** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIRO.EN2 set to '1'.

PFR24.1 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

Resource function is **INT1** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR0.EN1 set to '1'.

PFR24.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **INT0** input

Remark: This pin supports external interrupt wake up from STOP state if the PFR is set to '1' and interrupt request is enabled with ENIR0.EN0 set to '1'.

Remark: It is generally possible to use the External Interrupt Triggers INT also in the general purpose port input mode (PFR='0' and DDR='0'). In that case, wakeup from STOP state is not possible.

MB91460 Series**■ P25:** The functions of Port 25 are controlled by PFR25

	Addr	7	6	5	4	3	2	1	0	initial
PFR25	0D99h	PFR25.7	PFR25.6	PFR25.5	PFR25.4	PFR25.3	PFR25.2	PFR25.1	PFR25.0	0000 0000
EPFR25	0DD9h	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P25[7:0] is input/output for Stepper Motor PWM output signals and Comparator Inputs **SMC2M**, **SMC2P**, **SMC1M**, **SMC1P** of channels 4 and 5. Otherwise, the port can be used as general purpose port.

PFR25.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC2M5 output, CMP5 input if selected
PFR25.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC2P5 output, CMP5 input if selected
PFR25.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC1M5 output, CMP5 input if selected
PFR25.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC1P5 output, CMP5 input if selected
PFR25.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC2M4 output, CMP4 input if selected
PFR25.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC2P4 output, CMP4 input if selected
PFR25.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC1M4 output, CMP4 input if selected
PFR25.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SMC1P4 output, CMP4 input if selected

■ **P26:** The functions of Port 26 are controlled by PFR26 and EPFR26

	Addr	7	6	5	4	3	2	1	0	initial
PFR26	0D9Ah	PFR26.7	PFR26.6	PFR26.5	PFR26.4	PFR26.3	PFR26.2	PFR26.1	PFR26.0	0000 0000
EPFR26	0DDAh	EPFR26.7	EPFR26.6	EPFR26.5	EPFR26.4	EPFR26.3	EPFR26.2	EPFR26.1	EPFR26.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P26[7:0] is input/output for Stepper Motor PWM output signals and Comparator Inputs **SMC2M**, **SMC2P**, **SMC1M**, **SMC1P** of channels 2 and 3, and A/D converter analogue inputs **AN[31:24]**. Otherwise, the port can be used as general purpose port.

Additionally, P26[7:0] can be used for External Interrupt Triggers **INT[31:24]**, if the analog functions are disabled.

PFR26.7 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR26.7 0 - Resource function is **SMC2M3** output, CMP3 and/or AN31 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE15 1 - Resource function is **AN31** input
0 - Resource function is **INT31** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (**ADERH**='0') and interrupt request is enabled with **ENIR3.EN7** = '1'.

PFR26.6 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR26.6 0 - Resource function is **SMC2P3** output, CMP3 and/or AN30 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE14 1 - Resource function is **AN30** input
0 - Resource function is **INT30** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (**ADERH**='0') and interrupt request is enabled with **ENIR3.EN6** = '1'.

PFR26.5 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR26.5 0 - Resource function is **SMC1M3** output, CMP3 and/or AN29 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE13 1 - Resource function is **AN29** input
0 - Resource function is **INT29** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (**ADERH**='0') and interrupt request is enabled with **ENIR3.EN5** = '1'.

MB91460 Series

- PFR26.4 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
- EPFR26.4 0 - Resource function is **SMC1P3** output, CMP3 and/or AN28 input if selected
 1 - Resource function is ADC or External Interrupt:
- ADERH.ADE12 1 - Resource function is **AN28** input
 0 - Resource function is **INT28** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR3.EN4 = '1'.
- PFR26.3 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
- EPFR26.3 0 - Resource function is **SMC2M2** output, CMP2 and/or AN27 input if selected
 1 - Resource function is ADC or External Interrupt:
- ADERH.ADE11 1 - Resource function is **AN27** input
 0 - Resource function is **INT27** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR3.EN3 = '1'.
- PFR26.2 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
- EPFR26.2 0 - Resource function is **SMC2P2** output, CMP2 and/or AN26 input if selected
 1 - Resource function is ADC or External Interrupt:
- ADERH.ADE10 1 - Resource function is **AN26** input
 0 - Resource function is **INT26** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR3.EN2 = '1'.
- PFR26.1 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
- EPFR26.1 0 - Resource function is **SMC1M2** output, CMP2 and/or AN25 input if selected
 1 - Resource function is ADC or External Interrupt:
- ADERH.ADE9 1 - Resource function is **AN25** input
 0 - Resource function is **INT25** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR3.EN1 = '1'.

PFR26.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

EPFR26.0 0 - Resource function is **SMC1P2** output, CMP2 and/or AN24 input if selected
 1 - Resource function is ADC or External Interrupt:

ADERH.ADE8 1 - Resource function is **AN24** input
 0 - Resource function is **INT24** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR3.EN0 = '1'.

Remark: It is generally possible to use the External Interrupt Triggers INT also in the general purpose port input mode (PFR='0' and DDR='0'). In that case, wakeup from STOP state is not possible.

MB91460 Series**■ P27:** The functions of Port 27 are controlled by PFR27 and EPFR27

	Addr	7	6	5	4	3	2	1	0	initial
PFR27	0D9Bh	PFR27.7	PFR27.6	PFR27.5	PFR27.4	PFR27.3	PFR27.2	PFR27.1	PFR27.0	0000 0000
EPFR27	0DDBh	EPFR27.7	EPFR27.6	EPFR27.5	EPFR27.4	EPFR27.3	EPFR27.2	EPFR27.1	EPFR27.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P27[7:0] is input/output for Stepper Motor PWM output signals and Comparator Inputs **SMC2M**, **SMC2P**, **SMC1M**, **SMC1P** of channels 0 and 1, and A/D converter analogue inputs **AN[23:16]**. Otherwise, the port can be used as general purpose port.

PFR27.7 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR27.7 0 - Resource function is **SMC2M1** output, CMP1 and/or AN23 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE7 1 - Resource function is **AN23** input
0 - Resource function is **INT23** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN7 = '1'.

PFR27.6 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR27.6 0 - Resource function is **SMC2P1** output, CMP1 and/or AN22 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE6 1 - Resource function is **AN22** input
0 - Resource function is **INT22** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN6 = '1'.

PFR27.5 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:

EPFR27.5 0 - Resource function is **SMC1M1** output, CMP1 and/or AN21 input if selected
1 - Resource function is ADC or External Interrupt:

ADERH.ADE5 1 - Resource function is **AN21** input
0 - Resource function is **INT21** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN5 = '1'.

- PFR27.4 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:
- EPFR27.4 0 - Resource function is **SMC1P1** output, CMP1 and/or AN20 input if selected
1 - Resource function is ADC or External Interrupt:
- ADERH.ADE4 1 - Resource function is **AN20** input
0 - Resource function is **INT20** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN4 = '1'.
- PFR27.3 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:
- EPFR27.3 0 - Resource function is **SMC2M0** output, CMP0 and/or AN19 input if selected
1 - Resource function is ADC or External Interrupt:
- ADERH.ADE3 1 - Resource function is **AN19** input
0 - Resource function is **INT19** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN3 = '1'.
- PFR27.2 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:
- EPFR27.2 0 - Resource function is **SMC2P0** output, CMP0 and/or AN18 input if selected
1 - Resource function is ADC or External Interrupt:
- ADERH.ADE2 1 - Resource function is **AN18** input
0 - Resource function is **INT18** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN2 = '1'.
- PFR27.1 0 - Port is in general purpose port mode.
1 - Port is in resource function mode:
- EPFR27.1 0 - Resource function is **SMC1M0** output, CMP0 and/or AN17 input if selected
1 - Resource function is ADC or External Interrupt:
- ADERH.ADE1 1 - Resource function is **AN17** input
0 - Resource function is **INT17** input
- Remark:** This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN1 = '1'.

MB91460 Series

PFR27.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

EPFR27.0 0 - Resource function is **SMC1P0** output, CMP0 and/or AN16 input if selected
 1 - Resource function is ADC or External Interrupt:

ADERH.ADE0 1 - Resource function is **AN16** input
 0 - Resource function is **INT16** input

Remark: This pin supports external interrupt wake up from STOP state if PFR and EPFR are set to '1', the ADC channel is disabled (ADERH='0') and interrupt request is enabled with ENIR2.EN0 = '1'.

Remark: It is generally possible to use the External Interrupt Triggers INT also in the general purpose port input mode (PFR='0' and DDR='0'). In that case, wakeup from STOP state is not possible.

■ **P28:** The functions of Port 28 are controlled by PFR28

	Addr	7	6	5	4	3	2	1	0	initial
PFR28	0D9Ch	PFR28.7	PFR28.6	PFR28.5	PFR28.4	PFR28.3	PFR28.2	PFR28.1	PFR28.0	0000 0000
EPFR28	0DDCh	EPFR28.7	EPFR28.6	EPFR28.5	EPFR28.4	-	EPFR28.2	EPFR28.1	EPFR28.0	0000 -000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P28[7:0] is input/output for A/D converter analogue inputs **AN[15:8]**, and D/A converter analogue outputs **DA[1:0]** or **APIX[®]** sideband I/O. Otherwise, the port can be used as general purpose port.

- PFR28.7** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.7 0 - Resource function is **AN15** input and/or **DA1** output
 1 - Resource function is **APIX TEN0** input
 Remark: DAC output and ADC input can be enabled in parallel
- PFR28.6** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.6 0 - Resource function is **AN14** input and/or **DA0** output
 1 - Resource function is **APIX TDA01** output
 Remark: DAC output and ADC input can be enabled in parallel
- PFR28.5** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.5 0 - Resource function is **AN13** input
 1 - Resource function is **APIX TDA00** output
- PFR28.4** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.5 0 - Resource function is **AN12** input
 1 - Resource function is **APIX TCK0** output
- PFR28.3** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **AN11** input
- PFR28.2** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.2 0 - Resource function is **AN10** input
 1 - Resource function is **APIX RDA01** input
- PFR28.1** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.1 0 - Resource function is **AN9** input
 1 - Resource function is **APIX RDA00** input
- PFR28.0** 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 EPFR28.1 0 - Resource function is **AN8** input
 1 - Resource function is **APIX RCK0** input

MB91460 Series**■ P29:** The functions of Port 29 are controlled by PFR29

	Addr	7	6	5	4	3	2	1	0	initial
PFR29	0D9Dh	PFR29.7	PFR29.6	PFR29.5	PFR29.4	PFR29.3	PFR29.2	PFR29.1	PFR29.0	0000 0000
EPFR29	0DDh	-	-	-	-	-	-	-	-	----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P29[7:0] is input/output for A/D converter analogue inputs **AN[7:0]**. Otherwise, the port can be used as general purpose port.

PFR29.7 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN7** input

PFR29.6 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN6** input

PFR29.5 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN5** input

PFR29.4 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN4** input

PFR29.3 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN3** input

PFR29.2 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN2** input

PFR29.1 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN1** input

PFR29.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **AN0** input

■ **P30:** The functions of Port 30 are controlled by PFR30 and EPFR30

	Addr	7	6	5	4	3	2	1	0	initial
PFR30	0D9Eh	PFR30.7	PFR30.6	PFR30.5	PFR30.4	PFR30.3	PFR30.2	PFR30.1	PFR30.0	0000 0000
EPFR30	0DDEh	EPFR30.7	EPFR30.6	EPFR30.5	EPFR30.4	EPFR30.3	EPFR30.2	EPFR30.1	EPFR30.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P30[7:0] is input/output for LCD controller reference voltage analogue inputs **V[3:0]**, and LCD controller common driver outputs **COM[3:0]**, or it is output for the PPGs **PPG[23:16]**. Otherwise, the port can be used as general purpose port.

PFR30.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.7 0 - Resource function is V3 input 1 - Resource function is PPG23 output Resource function is V3 input
PFR30.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.6 0 - Resource function is V2 input 1 - Resource function is PPG22 output
PFR30.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.5 0 - Resource function is V1 input 1 - Resource function is PPG21 output
PFR30.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.4 0 - Resource function is V0 input 1 - Resource function is PPG20 output
PFR30.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.3 0 - Resource function is COM3 output 1 - Resource function is PPG19 output
PFR30.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.2 0 - Resource function is COM2 output 1 - Resource function is PPG18 output
PFR30.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.1 0 - Resource function is COM1 output 1 - Resource function is PPG17 output
PFR30.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR30.0 0 - Resource function is COM0 output 1 - Resource function is PPG16 output

MB91460 Series

■ **P31:** The functions of Port 31 are controlled by PFR31 and EPFR31

	Addr	7	6	5	4	3	2	1	0	initial
PFR31	0D9Fh	PFR31.7	PFR31.6	PFR31.5	PFR31.4	PFR31.3	PFR31.2	PFR31.1	PFR31.0	0000 0000
EPFR31	0DDFh	-	EPFR31.6	EPFR31.5	EPFR31.4	-	EPFR31.2	EPFR31.1	EPFR31.0	-000 -000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P31[7:0] is input/output for LCD controller segment driver outputs **SEG[39:32]** or input/output for FlexRay. Otherwise, the port can be used as general purpose port.

PFR31.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SEG39 output
PFR31.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.6 0 - Resource function is SEG38 output 1 - Resource function is RXDB input
PFR31.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.5 0 - Resource function is SEG37 output 1 - Resource function is TXENB output
PFR31.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.4 0 - Resource function is SEG36 output 1 - Resource function is TXDB output
PFR31.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SEG35 output
PFR31.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.2 0 - Resource function is SEG34 output 1 - Resource function is RXDA input
PFR31.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.1 0 - Resource function is SEG33 output 1 - Resource function is TXENA output
PFR31.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR31.0 0 - Resource function is SEG32 output 1 - Resource function is TXDA output

■ **P32:** The functions of Port 32 are controlled by PFR32 and EPFR32

	Addr	7	6	5	4	3	2	1	0	initial
PFR32	0DA0h	PFR32.7	PFR32.6	PFR32.5	PFR32.4	PFR32.3	PFR32.2	PFR32.1	PFR32.0	0000 0000
EPFR32	0DE0h	EPFR32.7	EPFR32.6	EPFR32.5	EPFR32.4	EPFR32.3	EPFR32.2	EPFR32.1	EPFR32.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P32[7:0] is input/output for LCD controller segment driver outputs **SEG[31:24]**, and LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 14 and 15, as well as output for **PPG[31:30]**. Otherwise, the port can be used as general purpose port.

PFR32.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.6 0 - Resource function is SEG31 output 1 - Resource function is PPG31 output
PFR32.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.6 0 - Resource function is SEG30 output 1 - Resource function is SCK15 input/output
PFR32.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.5 0 - Resource function is SEG29 output 1 - Resource function is SOT15 output
PFR32.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.4 0 - Resource function is SEG28 output 1 - Resource function is SIN15 input
PFR32.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.3 0 - Resource function is SEG27 output 1 - Resource function is PPG30 output
PFR32.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.2 0 - Resource function is SEG26 output 1 - Resource function is SCK14 input/output
PFR32.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.1 0 - Resource function is SEG25 output 1 - Resource function is SOT14 output
PFR32.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR32.0 0 - Resource function is SEG24 output 1 - Resource function is SIN14 input

MB91460 Series

■ **P33:** The functions of Port 33 are controlled by PFR33 and EPFR33

	Addr	7	6	5	4	3	2	1	0	initial
PFR33	0DA1h	PFR33.7	PFR33.6	PFR33.5	PFR33.4	PFR33.3	PFR33.2	PFR33.1	PFR33.0	0000 0000
EPFR33	0DE1h	EPFR33.7	EPFR33.6	EPFR33.5	EPFR33.4	EPFR33.3	EPFR33.2	EPFR33.1	EPFR33.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P33[7:0] is input/output for LCD controller segment driver outputs **SEG[23:16]**, and LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 12 and 13, as well as output for **PPG[29:28]**. Otherwise, the port can be used as general purpose port.

PFR33.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.7 0 - Resource function is SEG23 output 1 - Resource function is PPG29 output
PFR33.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.6 0 - Resource function is SEG22 output 1 - Resource function is SCK13 input/output
PFR33.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.5 0 - Resource function is SEG21 output 1 - Resource function is SOT13 output
PFR33.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.4 0 - Resource function is SEG20 output 1 - Resource function is SIN13 input
PFR33.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.3 0 - Resource function is SEG19 output 1 - Resource function is PPG28 output
PFR33.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.2 0 - Resource function is SEG18 output 1 - Resource function is SCK12 input/output
PFR33.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.1 0 - Resource function is SEG17 output 1 - Resource function is SOT12 output
PFR33.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR33.0 0 - Resource function is SEG16 output 1 - Resource function is SIN12 input

■ **P34:** The functions of Port 34 are controlled by PFR34 and EPFR34

	Addr	7	6	5	4	3	2	1	0	initial
PFR34	0DA2h	PFR34.7	PFR34.6	PFR34.5	PFR34.4	PFR34.3	PFR34.2	PFR34.1	PFR34.0	0000 0000
EPFR34	0DE2h	EPFR34.7	EPFR34.6	EPFR34.5	EPFR34.4	EPFR34.3	EPFR34.2	EPFR34.1	EPFR34.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P34[7:0] is input/output for LCD controller segment driver outputs **SEG[15:8]**, and LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 10 and 11, as well as output for **PPG[27:26]**. Otherwise, the port can be used as general purpose port.

PFR34.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.7 0 - Resource function is SEG15 output 1 - Resource function is PPG27 output
PFR34.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.6 0 - Resource function is SEG14 output 1 - Resource function is SCK11 input/output
PFR34.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.5 0 - Resource function is SEG13 output 1 - Resource function is SOT11 output
PFR34.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.4 0 - Resource function is SEG12 output 1 - Resource function is SIN11 input
PFR34.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.3 0 - Resource function is SEG11 output 1 - Resource function is PPG26 output
PFR34.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.2 0 - Resource function is SEG10 output 1 - Resource function is SCK10 input/output
PFR34.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.1 0 - Resource function is SEG9 output 1 - Resource function is SOT10 output
PFR34.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR34.0 0 - Resource function is SEG8 output 1 - Resource function is SIN10 input

MB91460 Series

■ **P35:** The functions of Port 35 are controlled by PFR35 and EPFR35

	Addr	7	6	5	4	3	2	1	0	initial
PFR35	0DA3h	PFR35.7	PFR35.6	PFR35.5	PFR35.4	PFR35.3	PFR35.2	PFR35.1	PFR35.0	0000 0000
EPFR35	0DE3h	EPFR35.7	EPFR35.6	EPFR35.5	EPFR35.4	EPFR35.3	EPFR35.2	EPFR35.1	EPFR35.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P35[7:0] is input/output for LCD controller segment driver outputs **SEG[7:0]**, and LIN-UART serial communication signals **SCK**, **SOT**, **SIN** of channels 8 and 9, as well as output for **PPG[25:24]**. Otherwise, the port can be used as general purpose port.

PFR35.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.7 0 - Resource function is SEG7 output 1 - Resource function is PPG25 output
PFR35.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.6 0 - Resource function is SEG6 output 1 - Resource function is SCK9 input/output
PFR35.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.5 0 - Resource function is SEG5 output 1 - Resource function is SOT9 output
PFR35.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.4 0 - Resource function is SEG4 output 1 - Resource function is SIN9 input
PFR35.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.3 0 - Resource function is SEG3 output 1 - Resource function is PPG24 output
PFR35.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.2 0 - Resource function is SEG2 output 1 - Resource function is SCK8 input/output
PFR35.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.1 0 - Resource function is SEG1 output 1 - Resource function is SOT8 output
PFR35.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: EPFR35.0 0 - Resource function is SEG0 output 1 - Resource function is SIN8 input

■ **P36:** The functions of Port 36 are controlled by PFR36

	Addr	7	6	5	4	3	2	1	0	initial
PFR36	0DA4h	PFR36.7	PFR36.6	PFR36.5	-	PFR36.3	PFR36.2	-	-	000- 00--
EPFR36	-	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P36[7:0] is input/output for **Media LB** and **I2S**. Otherwise, the port can be used as general purpose port.

- PFR36.7 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **MLBCLK** input
- PFR36.6 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **MLBSIG** input/output
- PFR36.5 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **MLBDAT** input/output
- PFR36.3 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **ISCK0** input/output
- PFR36.2 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:
 Resource function is **WS0** input/output

MB91460 Series

■ **P37:** The functions of Port 37 are not controlled by PFR / EPFR.

	Addr	7	6	5	4	3	2	1	0	initial
PFR37	-	-	-	-	-	-	-	-	-	-----
EPFR37	-	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Port 37 is a general purpose port and does not have resource functions.

■ **P38:** The functions of Port 38 are controlled by PFR38

	Addr	7	6	5	4	3	2	1	0	initial
PFR38	0DA6h	-	-	-	-	-	-	PFR38.1	PFR38.0	---- --00
EPFR38	-	-	-	-	-	-	-	-	-	-----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P38[1:0] is input/output for **I2S** sound data. Otherwise, the port can be used as general purpose port.

PFR38.1 0 - Port is in general purpose port mode.

 1 - Port is in resource function mode:

 Resource function is **SD9** input/output

PFR38.0 0 - Port is in general purpose port mode.

 1 - Port is in resource function mode:

 Resource function is **SD8** input/output

■ **P39:** The functions of Port 39 are controlled by PFR39

	Addr	7	6	5	4	3	2	1	0	initial
PFR39	0DA7h	PFR39.7	PFR39.6	PFR39.5	PFR39.4	PFR39.3	PFR39.2	PFR39.1	PFR39.0	0000 0000
EPFR39	-	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P39[7:0] is input/output for **I2S** sound data. Otherwise, the port can be used as general purpose port.

PFR39.7	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD7 input/output
PFR39.6	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD6 input/output
PFR39.5	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD5 input/output
PFR39.4	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD4 input/output
PFR39.3	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD3 input/output
PFR39.2	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD2 input/output
PFR39.1	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD1 input/output
PFR39.0	0 - Port is in general purpose port mode. 1 - Port is in resource function mode: Resource function is SD0 input/output

MB91460 Series**■ P40:** The functions of Port 40 are controlled by PFR40

	Addr	7	6	5	4	3	2	1	0	initial
PFR40	0DA8h	PFR40.7	PFR40.6	PFR40.5	PFR40.4	PFR40.3	PFR40.2	PFR40.1	PFR40.0	0000 0000
EPFR40	-	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

P40[7:0] is input/output for I2C serial communication signals **SCL**, **SDA** of channels 4 to 7. Otherwise, the port can be used as general purpose port.

PFR40.7 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SCL7** open drain

PFR40.6 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SDA7** open drain

PFR40.5 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SCL6** open drain

PFR40.4 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SDA6** open drain

PFR40.3 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SCL5** open drain

PFR40.2 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SDA5** open drain

PFR40.1 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SCL4** open drain

PFR40.0 0 - Port is in general purpose port mode.
 1 - Port is in resource function mode:

Resource function is **SDA4** open drain

30.4.5 Port Input Level Selection

The input levels of each port can be programmed bit-wise between CMOS Hysteresis type A and B, Automotive Hysteresis and TTL level.

CMOS Hysteresis type A:	$V_{IL}=0.3 \times V_{DD}$	$V_{IH}=0.7 \times V_{DD}$
CMOS Hysteresis type B:	$V_{IL}=0.2 \times V_{DD}$	$V_{IH}=0.8 \times V_{DD}$
Automotive Hysteresis:	$V_{IL}=0.5 \times V_{DD}$	$V_{IH}=0.8 \times V_{DD}$
TTL	$V_{IL}=0.8 \text{ V}$	$V_{IH}=2.1 \text{ V}$
CMOS	$V_{IL}=0.3 \times V_{DD}$	$V_{IH}=0.7 \times V_{DD}$
Media LB	$V_{IL}=0.7 \text{ V}$	$V_{IH}=1.7 \text{ V}$

For setup, the Port Input Level Registers (PILR, EPILR) of each port are used.

Table 30.4-2 PILR settings MB91Fxxx and MB91FV460B (GP00 - GP35, GP37, GP40)

PILRx.y	EPILRx.y	Port Input Level
0 (default)	0 (default)	CMOS Hysteresis A
1	0	Automotive Hysteresis
0	1	TTL
1	1	CMOS Hysteresis B

Table 30.4-3 PILR settings MB91Fxxx and MB91FV460B (GP36[7:5])

PILRx.y	EPILRx.y	Port Input Level
0 (default)	0 (default)	CMOS
1	0	setting prohibited
0	1	Media LB
1	1	CMOS Hysteresis B

Table 30.4-4 PILR settings of MB91Fxxx and MB91FV460B (GP36[4:0], GP38, GP39)

PILRx.y	EPILRx.y ¹	Port Input Level
0 (default)	n.a.	CMOS
1		CMOS Hysteresis B

1. EPILR not implemented

MB91460 Series**Table 30.4-5 PILR settings of the MB91V460A (GP00 - GP13)**

PILRx.y	EPILRx.y	Port Input Level
0 (default)	0 (default)	CMOS Hysteresis A
1	0	Automotive Hysteresis
0	1	TTL
1	1	n.a.

Table 30.4-6 PILR settings of the MB91V460A (GP14 - GP35)

PILRx.y	EPILRx.y ¹	Port Input Level
0 (default)	n.a.	CMOS Hysteresis A
1		Automotive Hysteresis

1. EPILR14-EPILR40 are not implemented on MB91V460A.

Table 30.4-7 PILR register list

	Addr	7	6	5	4	3	2	1	0	initial
PILR00	0E40h	PILR00.7	PILR00.6	PILR00.5	PILR00.4	PILR00.3	PILR00.2	PILR00.1	PILR00.0	0000 0000
PILR01	0E41h	PILR01.7	PILR01.6	PILR01.5	PILR01.4	PILR01.3	PILR01.2	PILR01.1	PILR01.0	0000 0000
PILR02	0E42h	PILR02.7	PILR02.6	PILR02.5	PILR02.4	PILR02.3	PILR02.2	PILR02.1	PILR02.0	0000 0000
PILR03	0E43h	PILR03.7	PILR03.6	PILR03.5	PILR03.4	PILR03.3	PILR03.2	PILR03.1	PILR03.0	0000 0000
PILR04	0E44h	PILR04.7	PILR04.6	PILR04.5	PILR04.4	PILR04.3	PILR04.2	PILR04.1	PILR04.0	0000 0000
PILR05	0E45h	PILR05.7	PILR05.6	PILR05.5	PILR05.4	PILR05.3	PILR05.2	PILR05.1	PILR05.0	0000 0000
PILR06	0E46h	PILR06.7	PILR06.6	PILR06.5	PILR06.4	PILR06.3	PILR06.2	PILR06.1	PILR06.0	0000 0000
PILR07	0E47h	PILR07.7	PILR07.6	PILR07.5	PILR07.4	PILR07.3	PILR07.2	PILR07.1	PILR07.0	0000 0000
PILR08	0E48h	PILR08.7	PILR08.6	PILR08.5	PILR08.4	PILR08.3	PILR08.2	PILR08.1	PILR08.0	0000 0000
PILR09	0E49h	PILR09.7	PILR09.6	PILR09.5	PILR09.4	PILR09.3	PILR09.2	PILR09.1	PILR09.0	0000 0000
PILR10	0E4Ah	PILR10.7	PILR10.6	PILR10.5	PILR10.4	PILR10.3	PILR10.2	PILR10.1	PILR10.0	0000 0000
PILR11	0E4Bh	PILR11.7	PILR11.6	PILR11.5	PILR11.4	PILR11.3	PILR11.2	PILR11.1	PILR11.0	0000 0000
PILR12	0E4Ch	PILR12.7	PILR12.6	PILR12.5	PILR12.4	PILR12.3	PILR12.2	PILR12.1	PILR12.0	0000 0000
PILR13	0E4Dh	PILR13.7	PILR13.6	PILR13.5	PILR13.4	PILR13.3	PILR13.2	PILR13.1	PILR13.0	0000 0000
PILR14	0E4Eh	PILR14.7	PILR14.6	PILR14.5	PILR14.4	PILR14.3	PILR14.2	PILR14.1	PILR14.0	0000 0000
PILR15	0E4Fh	PILR15.7	PILR15.6	PILR15.5	PILR15.4	PILR15.3	PILR15.2	PILR15.1	PILR15.0	0000 0000
PILR16	0E50h	PILR16.7	PILR16.6	PILR16.5	PILR16.4	PILR16.3	PILR16.2	PILR16.1	PILR16.0	0000 0000
PILR17	0E51h	PILR17.7	PILR17.6	PILR17.5	PILR17.4	PILR17.3	PILR17.2	PILR17.1	PILR17.0	0000 0000
PILR18	0E52h	PILR18.7	PILR18.6	PILR18.5	PILR18.4	PILR18.3	PILR18.2	PILR18.1	PILR18.0	0000 0000
PILR19	0E53h	PILR19.7	PILR19.6	PILR19.5	PILR19.4	PILR19.3	PILR19.2	PILR19.1	PILR19.0	0000 0000

PILR20	0E54h	PILR20.7	PILR20.6	PILR20.5	PILR20.4	PILR20.3	PILR20.2	PILR20.1	PILR20.0	0000 0000
PILR21	0E55h	PILR21.7	PILR21.6	PILR21.5	PILR21.4	PILR21.3	PILR21.2	PILR21.1	PILR21.0	0000 0000
PILR22	0E56h	PILR22.7	PILR22.6	PILR22.5	PILR22.4	PILR22.3	PILR22.2	PILR22.1	PILR22.0	0000 0000
PILR23	0E57h	PILR23.7	PILR23.6	PILR23.5	PILR23.4	PILR23.3	PILR23.2	PILR23.1	PILR23.0	0000 0000
PILR24	0E58h	PILR24.7	PILR24.6	PILR24.5	PILR24.4	PILR24.3	PILR24.2	PILR24.1	PILR24.0	0000 0000
PILR25	0E59h	PILR25.7	PILR25.6	PILR25.5	PILR25.4	PILR25.3	PILR25.2	PILR25.1	PILR25.0	0000 0000
PILR26	0E5Ah	PILR26.7	PILR26.6	PILR26.5	PILR26.4	PILR26.3	PILR26.2	PILR26.1	PILR26.0	0000 0000
PILR27	0E5Bh	PILR27.7	PILR27.6	PILR27.5	PILR27.4	PILR27.3	PILR27.2	PILR27.1	PILR27.0	0000 0000
PILR28	0E5Ch	PILR28.7	PILR28.6	PILR28.5	PILR28.4	PILR28.3	PILR28.2	PILR28.1	PILR28.0	0000 0000
PILR29	0E5Dh	PILR29.7	PILR29.6	PILR29.5	PILR29.4	PILR29.3	PILR29.2	PILR29.1	PILR29.0	0000 0000
PILR30	0E5Eh	PILR30.7	PILR30.6	PILR30.5	PILR30.4	PILR30.3	PILR30.2	PILR30.1	PILR30.0	0000 0000
PILR31	0E5Fh	PILR31.7	PILR31.6	PILR31.5	PILR31.4	PILR31.3	PILR31.2	PILR31.1	PILR31.0	0000 0000
PILR32	0E60h	PILR32.7	PILR32.6	PILR32.5	PILR32.4	PILR32.3	PILR32.2	PILR32.1	PILR32.0	0000 0000
PILR33	0E61h	PILR33.7	PILR33.6	PILR33.5	PILR33.4	PILR33.3	PILR33.2	PILR33.1	PILR33.0	0000 0000
PILR34	0E62h	PILR34.7	PILR34.6	PILR34.5	PILR34.4	PILR34.3	PILR34.2	PILR34.1	PILR34.0	0000 0000
PILR35	0E63h	PILR35.7	PILR35.6	PILR35.5	PILR35.4	PILR35.3	PILR35.2	PILR35.1	PILR35.0	0000 0000
PILR36	0E64h	PILR36.7	PILR36.6	PILR36.5	PILR36.4	PILR36.3	PILR36.2	PILR36.1	PILR36.0	0000 0000
PILR37	0E65h	PILR37.7	PILR37.6	PILR37.5	PILR37.4	PILR37.3	PILR37.2	PILR37.1	PILR37.0	0000 0000
PILR38	0E66h	PILR38.7	PILR38.6	PILR38.5	PILR38.4	PILR38.3	PILR38.2	PILR38.1	PILR38.0	0000 0000
PILR39	0E67h	PILR39.7	PILR39.6	PILR39.5	PILR39.4	PILR39.3	PILR39.2	PILR39.1	PILR39.0	0000 0000
PILR40	0E68h	PILR40.7	PILR40.6	PILR40.5	PILR40.4	PILR40.3	PILR40.2	PILR40.1	PILR40.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 30.4-8 EPILR register list

	Addr	7	6	5	4	3	2	1	0	initial
EPILR00	0E80h	EPILR00.7	EPILR00.6	EPILR00.5	EPILR00.4	EPILR00.3	EPILR00.2	EPILR00.1	EPILR00.0	0000 0000
EPILR01	0E81h	EPILR01.7	EPILR01.6	EPILR01.5	EPILR01.4	EPILR01.3	EPILR01.2	EPILR01.1	EPILR01.0	0000 0000
EPILR02	0E82h	EPILR02.7	EPILR02.6	EPILR02.5	EPILR02.4	EPILR02.3	EPILR02.2	EPILR02.1	EPILR02.0	0000 0000
EPILR03	0E83h	EPILR03.7	EPILR03.6	EPILR03.5	EPILR03.4	EPILR03.3	EPILR03.2	EPILR03.1	EPILR03.0	0000 0000
EPILR04	0E84h	EPILR04.7	EPILR04.6	EPILR04.5	EPILR04.4	EPILR04.3	EPILR04.2	EPILR04.1	EPILR04.0	0000 0000
EPILR05	0E85h	EPILR05.7	EPILR05.6	EPILR05.5	EPILR05.4	EPILR05.3	EPILR05.2	EPILR05.1	EPILR05.0	0000 0000
EPILR06	0E86h	EPILR06.7	EPILR06.6	EPILR06.5	EPILR06.4	EPILR06.3	EPILR06.2	EPILR06.1	EPILR06.0	0000 0000
EPILR07	0E87h	EPILR07.7	EPILR07.6	EPILR07.5	EPILR07.4	EPILR07.3	EPILR07.2	EPILR07.1	EPILR07.0	0000 0000
EPILR08	0E88h	EPILR08.7	EPILR08.6	EPILR08.5	EPILR08.4	EPILR08.3	EPILR08.2	EPILR08.1	EPILR08.0	0000 0000
EPILR09	0E89h	EPILR09.7	EPILR09.6	EPILR09.5	EPILR09.4	EPILR09.3	EPILR09.2	EPILR09.1	EPILR09.0	0000 0000

MB91460 Series

EPILR10	0E8Ah	EPILR10.7	EPILR10.6	EPILR10.5	EPILR10.4	EPILR10.3	EPILR10.2	EPILR10.1	EPILR10.0	0000 0000
EPILR11	0E8Bh	EPILR11.7	EPILR11.6	EPILR11.5	EPILR11.4	EPILR11.3	EPILR11.2	EPILR11.1	EPILR11.0	0000 0000
EPILR12	0E8Ch	EPILR12.7	EPILR12.6	EPILR12.5	EPILR12.4	EPILR12.3	EPILR12.2	EPILR12.1	EPILR12.0	0000 0000
EPILR13	0E8Dh	EPILR13.7	EPILR13.6	EPILR13.5	EPILR13.4	EPILR13.3	EPILR13.2	EPILR13.1	EPILR13.0	0000 0000
EPILR14	0E8Eh	EPILR14.7	EPILR14.6	EPILR14.5	EPILR14.4	EPILR14.3	EPILR14.2	EPILR14.1	EPILR14.0	0000 0000
EPILR15	0E8Fh	EPILR15.7	EPILR15.6	EPILR15.5	EPILR15.4	EPILR15.3	EPILR15.2	EPILR15.1	EPILR15.0	0000 0000
EPILR16	0E90h	EPILR16.7	EPILR16.6	EPILR16.5	EPILR16.4	EPILR16.3	EPILR16.2	EPILR16.1	EPILR16.0	0000 0000
EPILR17	0E91h	EPILR17.7	EPILR17.6	EPILR17.5	EPILR17.4	EPILR17.3	EPILR17.2	EPILR17.1	EPILR17.0	0000 0000
EPILR18	0E92h	EPILR18.7	EPILR18.6	EPILR18.5	EPILR18.4	EPILR18.3	EPILR18.2	EPILR18.1	EPILR18.0	0000 0000
EPILR19	0E93h	EPILR19.7	EPILR19.6	EPILR19.5	EPILR19.4	EPILR19.3	EPILR19.2	EPILR19.1	EPILR19.0	0000 0000
EPILR20	0E94h	EPILR20.7	EPILR20.6	EPILR20.5	EPILR20.4	EPILR20.3	EPILR20.2	EPILR20.1	EPILR20.0	0000 0000
EPILR21	0E95h	EPILR21.7	EPILR21.6	EPILR21.5	EPILR21.4	EPILR21.3	EPILR21.2	EPILR21.1	EPILR21.0	0000 0000
EPILR22	0E96h	EPILR22.7	EPILR22.6	EPILR22.5	EPILR22.4	EPILR22.3	EPILR22.2	EPILR22.1	EPILR22.0	0000 0000
EPILR23	0E97h	EPILR23.7	EPILR23.6	EPILR23.5	EPILR23.4	EPILR23.3	EPILR23.2	EPILR23.1	EPILR23.0	0000 0000
EPILR24	0E98h	EPILR24.7	EPILR24.6	EPILR24.5	EPILR24.4	EPILR24.3	EPILR24.2	EPILR24.1	EPILR24.0	0000 0000
EPILR25	0E99h	EPILR25.7	EPILR25.6	EPILR25.5	EPILR25.4	EPILR25.3	EPILR25.2	EPILR25.1	EPILR25.0	0000 0000
EPILR26	0E9Ah	EPILR26.7	EPILR26.6	EPILR26.5	EPILR26.4	EPILR26.3	EPILR26.2	EPILR26.1	EPILR26.0	0000 0000
EPILR27	0E9Bh	EPILR27.7	EPILR27.6	EPILR27.5	EPILR27.4	EPILR27.3	EPILR27.2	EPILR27.1	EPILR27.0	0000 0000
EPILR28	0E9Ch	EPILR28.7	EPILR28.6	EPILR28.5	EPILR28.4	EPILR28.3	EPILR28.2	EPILR28.1	EPILR28.0	0000 0000
EPILR29	0E9Dh	EPILR29.7	EPILR29.6	EPILR29.5	EPILR29.4	EPILR29.3	EPILR29.2	EPILR29.1	EPILR29.0	0000 0000
EPILR30	0E9Eh	EPILR30.7	EPILR30.6	EPILR30.5	EPILR30.4	EPILR30.3	EPILR30.2	EPILR30.1	EPILR30.0	0000 0000
EPILR31	0E9Fh	EPILR31.7	EPILR31.6	EPILR31.5	EPILR31.4	EPILR31.3	EPILR31.2	EPILR31.1	EPILR31.0	0000 0000
EPILR32	0EA0h	EPILR32.7	EPILR32.6	EPILR32.5	EPILR32.4	EPILR32.3	EPILR32.2	EPILR32.1	EPILR32.0	0000 0000
EPILR33	0EA1h	EPILR33.7	EPILR33.6	EPILR33.5	EPILR33.4	EPILR33.3	EPILR33.2	EPILR33.1	EPILR33.0	0000 0000
EPILR34	0EA2h	EPILR34.7	EPILR34.6	EPILR34.5	EPILR34.4	EPILR34.3	EPILR34.2	EPILR34.1	EPILR34.0	0000 0000
EPILR35	0EA3h	EPILR35.7	EPILR35.6	EPILR35.5	EPILR35.4	EPILR35.3	EPILR35.2	EPILR35.1	EPILR35.0	0000 0000
EPILR36	0EA4h	EPILR36.7	EPILR36.6	EPILR36.5	-	-	-	-	-	000- ----
EPILR37	0EA5h	EPILR37.7	EPILR37.6	EPILR37.5	EPILR37.4	EPILR37.3	EPILR37.2	EPILR37.1	EPILR37.0	0000 0000
EPILR38	-	-	-	-	-	-	-	-	-	---- ----
EPILR39	-	-	-	-	-	-	-	-	-	---- ----
EPILR40	0EA8h	EPILR40.7	EPILR40.6	EPILR40.5	EPILR40.4	EPILR40.3	EPILR40.2	EPILR40.1	EPILR40.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

30.4.6 Programmable Pull-Up/Pull Down Resistors

The Ports listed in the following table have 50 kOhm Pull-Up and Pull-Down resistors, which can be enabled bit-wise. The function is enabled by the Port Pull Enable Registers (PPER) and controlled by the Port Pull Control Register (PPCR). The PPCR selects Pull-Up or Pull-Down. The Pull-Up/Pull-Down are disabled automatically in the STOP-HiZ state (STCR:STOP and STCR:HIZ set).

Note: On MB91F467D, the PU/PD are NOT disabled automatically in STOP-HiZ.

Bit	Port Pull-Up/Pull-Down Enable Registers	
	0 (default)	1
PPERx.y	Pull-Up/Pull-Down disabled	Pull-Up/Pull-Down enabled

Table 30.4-9 PPER register list

	Addr	7	6	5	4	3	2	1	0	initial
PPER00	0EC0h	PPER00.7	PPER00.6	PPER00.5	PPER00.4	PPER00.3	PPER00.2	PPER00.1	PPER00.0	0000 0000
PPER01	0EC1h	PPER01.7	PPER01.6	PPER01.5	PPER01.4	PPER01.3	PPER01.2	PPER01.1	PPER01.0	0000 0000
PPER02	0EC2h	PPER02.7	PPER02.6	PPER02.5	PPER02.4	PPER02.3	PPER02.2	PPER02.1	PPER02.0	0000 0000
PPER03	0EC3h	PPER03.7	PPER03.6	PPER03.5	PPER03.4	PPER03.3	PPER03.2	PPER03.1	PPER03.0	0000 0000
PPER04	0EC4h	PPER04.7	PPER04.6	PPER04.5	PPER04.4	PPER04.3	PPER04.2	PPER04.1	PPER04.0	0000 0000
PPER05	0EC5h	PPER05.7	PPER05.6	PPER05.5	PPER05.4	PPER05.3	PPER05.2	PPER05.1	PPER05.0	0000 0000
PPER06	0EC6h	PPER06.7	PPER06.6	PPER06.5	PPER06.4	PPER06.3	PPER06.2	PPER06.1	PPER06.0	0000 0000
PPER07	0EC7h	PPER07.7	PPER07.6	PPER07.5	PPER07.4	PPER07.3	PPER07.2	PPER07.1	PPER07.0	0000 0000
PPER08	0EC8h	PPER08.7	PPER08.6	PPER08.5	PPER08.4	PPER08.3	PPER08.2	PPER08.1	PPER08.0	0000 0000
PPER09	0EC9h	PPER09.7	PPER09.6	PPER09.5	PPER09.4	PPER09.3	PPER09.2	PPER09.1	PPER09.0	0000 0000
PPER10	0ECAh	PPER10.7	PPER10.6	PPER10.5	PPER10.4	PPER10.3	PPER10.2	PPER10.1	PPER10.0	0000 0000
PPER11	0ECBh	PPER11.7	PPER11.6	PPER11.5	PPER11.4	PPER11.3	PPER11.2	PPER11.1	PPER11.0	0000 0000
PPER12	0ECCh	PPER12.7	PPER12.6	PPER12.5	PPER12.4	PPER12.3	PPER12.2	PPER12.1	PPER12.0	0000 0000
PPER13	0ECDh	PPER13.7	PPER13.6	PPER13.5	PPER13.4	PPER13.3	PPER13.2	PPER13.1	PPER13.0	0000 0000
PPER14	0ECEh	PPER14.7	PPER14.6	PPER14.5	PPER14.4	PPER14.3	PPER14.2	PPER14.1	PPER14.0	0000 0000
PPER15	0ECFh	PPER15.7	PPER15.6	PPER15.5	PPER15.4	PPER15.3	PPER15.2	PPER15.1	PPER15.0	0000 0000
PPER16	0ED0h	PPER16.7	PPER16.6	PPER16.5	PPER16.4	PPER16.3	PPER16.2	PPER16.1	PPER16.0	0000 0000
PPER17	0ED1h	PPER17.7	PPER17.6	PPER17.5	PPER17.4	PPER17.3	PPER17.2	PPER17.1	PPER17.0	0000 0000
PPER18	0ED2h	PPER18.7	PPER18.6	PPER18.5	PPER18.4	PPER18.3	PPER18.2	PPER18.1	PPER18.0	0000 0000
PPER19	0ED3h	PPER19.7	PPER19.6	PPER19.5	PPER19.4	PPER19.3	PPER19.2	PPER19.1	PPER19.0	0000 0000
PPER20	0ED4h	PPER20.7	PPER20.6	PPER20.5	PPER20.4	PPER20.3	PPER20.2	PPER20.1	PPER20.0	0000 0000
PPER21	0ED5h	PPER21.7	PPER21.6	PPER21.5	PPER21.4	PPER21.3	PPER21.2	PPER21.1	PPER21.0	0000 0000
PPER22	0ED6h	PPER22.7	PPER22.6	PPER22.5	PPER22.4	PPER22.3	PPER22.2	PPER22.1	PPER22.0	0000 0000
PPER23	0ED7h	PPER23.7	PPER23.6	PPER23.5	PPER23.4	PPER23.3	PPER23.2	PPER23.1	PPER23.0	0000 0000

PPER24	0ED8h	PPER24.7	PPER24.6	PPER24.5	PPER24.4	PPER24.3	PPER24.2	PPER24.1	PPER24.0	0000 0000
PPER25	0ED9h	PPER25.7	PPER25.6	PPER25.5	PPER25.4	PPER25.3	PPER25.2	PPER25.1	PPER25.0	0000 0000
PPER26	0EDAh	PPER26.7	PPER26.6	PPER26.5	PPER26.4	PPER26.3	PPER26.2	PPER26.1	PPER26.0	0000 0000
PPER27	0EDBh	PPER27.7	PPER27.6	PPER27.5	PPER27.4	PPER27.3	PPER27.2	PPER27.1	PPER27.0	0000 0000
PPER28	0EDCh	PPER28.7	PPER28.6	PPER28.5	PPER28.4	PPER28.3	PPER28.2	PPER28.1	PPER28.0	0000 0000
PPER29	0EDDh	PPER29.7	PPER29.6	PPER29.5	PPER29.4	PPER29.3	PPER29.2	PPER29.1	PPER29.0	0000 0000
PPER30	0EDEh	PPER30.7	PPER30.6	PPER30.5	PPER30.4	PPER30.3	PPER30.2	PPER30.1	PPER30.0	0000 0000
PPER31	0EDFh	PPER31.7	PPER31.6	PPER31.5	PPER31.4	PPER31.3	PPER31.2	PPER31.1	PPER31.0	0000 0000
PPER32	0EE0h	PPER32.7	PPER32.6	PPER32.5	PPER32.4	PPER32.3	PPER32.2	PPER32.1	PPER32.0	0000 0000
PPER33	0EE1h	PPER33.7	PPER33.6	PPER33.5	PPER33.4	PPER33.3	PPER33.2	PPER33.1	PPER33.0	0000 0000
PPER34	0EE2h	PPER34.7	PPER34.6	PPER34.5	PPER34.4	PPER34.3	PPER34.2	PPER34.1	PPER34.0	0000 0000
PPER35	0EE3h	PPER35.7	PPER35.6	PPER35.5	PPER35.4	PPER35.3	PPER35.2	PPER35.1	PPER35.0	0000 0000
PPER36	0EE4h	PPER36.7	PPER36.6	PPER36.5	PPER36.4	PPER36.3	PPER36.2	PPER36.1	PPER36.0	0000 0000
PPER37	0EE5h	PPER37.7	PPER37.6	PPER37.5	PPER37.4	PPER37.3	PPER37.2	PPER37.1	PPER37.0	0000 0000
PPER38	0EE6h	PPER38.7	PPER38.6	PPER38.5	PPER38.4	PPER38.3	PPER38.2	PPER38.1	PPER38.0	0000 0000
PPER39	0EE7h	PPER39.7	PPER39.6	PPER39.5	PPER39.4	PPER39.3	PPER39.2	PPER39.1	PPER39.0	0000 0000
PPER40	0EE8h	PPER40.7	PPER40.6	PPER40.5	PPER40.4	PPER40.3	PPER40.2	PPER40.1	PPER40.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

It is necessary to disable the Pull-Up/Pull Downs first ([See "Programmable Pull-Up/Pull Down Resistors" on P. 650](#)), before changing the setting of the PPCRx.y bit. After changing this bit you have to reenale the Pull-Up/Pull Downs.

Bit	Port Pull-Up/Pull-Down Control Registers	
	0	1 (default)
PPCRx.y	Pull Down is selected	Pull-Up is selected

Table 30.4-10 PPCR register list

	Addr	7	6	5	4	3	2	1	0	initial
PPCR00	0F00h	PPCR00.7	PPCR00.6	PPCR00.5	PPCR00.4	PPCR00.3	PPCR00.2	PPCR00.1	PPCR00.0	1111 1111
PPCR01	0F01h	PPCR01.7	PPCR01.6	PPCR01.5	PPCR01.4	PPCR01.3	PPCR01.2	PPCR01.1	PPCR01.0	1111 1111
PPCR02	0F02h	PPCR02.7	PPCR02.6	PPCR02.5	PPCR02.4	PPCR02.3	PPCR02.2	PPCR02.1	PPCR02.0	1111 1111
PPCR03	0F03h	PPCR03.7	PPCR03.6	PPCR03.5	PPCR03.4	PPCR03.3	PPCR03.2	PPCR03.1	PPCR03.0	1111 1111
PPCR04	0F04h	PPCR04.7	PPCR04.6	PPCR04.5	PPCR04.4	PPCR04.3	PPCR04.2	PPCR04.1	PPCR04.0	1111 1111
PPCR05	0F05h	PPCR05.7	PPCR05.6	PPCR05.5	PPCR05.4	PPCR05.3	PPCR05.2	PPCR05.1	PPCR05.0	1111 1111
PPCR06	0F06h	PPCR06.7	PPCR06.6	PPCR06.5	PPCR06.4	PPCR06.3	PPCR06.2	PPCR06.1	PPCR06.0	1111 1111
PPCR07	0F07h	PPCR07.7	PPCR07.6	PPCR07.5	PPCR07.4	PPCR07.3	PPCR07.2	PPCR07.1	PPCR07.0	1111 1111

PPCR08	0F08h	PPCR08.7	PPCR08.6	PPCR08.5	PPCR08.4	PPCR08.3	PPCR08.2	PPCR08.1	PPCR08.0	1111 1111
PPCR09	0F09h	PPCR09.7	PPCR09.6	PPCR09.5	PPCR09.4	PPCR09.3	PPCR09.2	PPCR09.1	PPCR09.0	1111 1111
PPCR10	0F0Ah	PPCR10.7	PPCR10.6	PPCR10.5	PPCR10.4	PPCR10.3	PPCR10.2	PPCR10.1	PPCR10.0	1111 1111
PPCR11	0F0Bh	PPCR11.7	PPCR11.6	PPCR11.5	PPCR11.4	PPCR11.3	PPCR11.2	PPCR11.1	PPCR11.0	1111 1111
PPCR12	0F0Ch	PPCR12.7	PPCR12.6	PPCR12.5	PPCR12.4	PPCR12.3	PPCR12.2	PPCR12.1	PPCR12.0	1111 1111
PPCR13	0F0Dh	PPCR13.7	PPCR13.6	PPCR13.5	PPCR13.4	PPCR13.3	PPCR13.2	PPCR13.1	PPCR13.0	1111 1111
PPCR14	0F0Eh	PPCR14.7	PPCR14.6	PPCR14.5	PPCR14.4	PPCR14.3	PPCR14.2	PPCR14.1	PPCR14.0	1111 1111
PPCR15	0F0Fh	PPCR15.7	PPCR15.6	PPCR15.5	PPCR15.4	PPCR15.3	PPCR15.2	PPCR15.1	PPCR15.0	1111 1111
PPCR16	0F10h	PPCR16.7	PPCR16.6	PPCR16.5	PPCR16.4	PPCR16.3	PPCR16.2	PPCR16.1	PPCR16.0	1111 1111
PPCR17	0F11h	PPCR17.7	PPCR17.6	PPCR17.5	PPCR17.4	PPCR17.3	PPCR17.2	PPCR17.1	PPCR17.0	1111 1111
PPCR18	0F12h	PPCR18.7	PPCR18.6	PPCR18.5	PPCR18.4	PPCR18.3	PPCR18.2	PPCR18.1	PPCR18.0	1111 1111
PPCR19	0F13h	PPCR19.7	PPCR19.6	PPCR19.5	PPCR19.4	PPCR19.3	PPCR19.2	PPCR19.1	PPCR19.0	1111 1111
PPCR20	0F14h	PPCR20.7	PPCR20.6	PPCR20.5	PPCR20.4	PPCR20.3	PPCR20.2	PPCR20.1	PPCR20.0	1111 1111
PPCR21	0F15h	PPCR21.7	PPCR21.6	PPCR21.5	PPCR21.4	PPCR21.3	PPCR21.2	PPCR21.1	PPCR21.0	1111 1111
PPCR22	0F16h	PPCR22.7	PPCR22.6	PPCR22.5	PPCR22.4	PPCR22.3	PPCR22.2	PPCR22.1	PPCR22.0	1111 1111
PPCR23	0F17h	PPCR23.7	PPCR23.6	PPCR23.5	PPCR23.4	PPCR23.3	PPCR23.2	PPCR23.1	PPCR23.0	1111 1111
PPCR24	0F18h	PPCR24.7	PPCR24.6	PPCR24.5	PPCR24.4	PPCR24.3	PPCR24.2	PPCR24.1	PPCR24.0	1111 1111
PPCR25	0F19h	PPCR25.7	PPCR25.6	PPCR25.5	PPCR25.4	PPCR25.3	PPCR25.2	PPCR25.1	PPCR25.0	1111 1111
PPCR26	0F1Ah	PPCR26.7	PPCR26.6	PPCR26.5	PPCR26.4	PPCR26.3	PPCR26.2	PPCR26.1	PPCR26.0	1111 1111
PPCR27	0F1Bh	PPCR27.7	PPCR27.6	PPCR27.5	PPCR27.4	PPCR27.3	PPCR27.2	PPCR27.1	PPCR27.0	1111 1111
PPCR28	0F1Ch	PPCR28.7	PPCR28.6	PPCR28.5	PPCR28.4	PPCR28.3	PPCR28.2	PPCR28.1	PPCR28.0	1111 1111
PPCR29	0F1Dh	PPCR29.7	PPCR29.6	PPCR29.5	PPCR29.4	PPCR29.3	PPCR29.2	PPCR29.1	PPCR29.0	1111 1111
PPCR30	0F1Eh	PPCR30.7	PPCR30.6	PPCR30.5	PPCR30.4	PPCR30.3	PPCR30.2	PPCR30.1	PPCR30.0	1111 1111
PPCR31	0F1Fh	PPCR31.7	PPCR31.6	PPCR31.5	PPCR31.4	PPCR31.3	PPCR31.2	PPCR31.1	PPCR31.0	1111 1111
PPCR32	0F20h	PPCR32.7	PPCR32.6	PPCR32.5	PPCR32.4	PPCR32.3	PPCR32.2	PPCR32.1	PPCR32.0	1111 1111
PPCR33	0F21h	PPCR33.7	PPCR33.6	PPCR33.5	PPCR33.4	PPCR33.3	PPCR33.2	PPCR33.1	PPCR33.0	1111 1111
PPCR34	0F22h	PPCR34.7	PPCR34.6	PPCR34.5	PPCR34.4	PPCR34.3	PPCR34.2	PPCR34.1	PPCR34.0	1111 1111
PPCR35	0F23h	PPCR35.7	PPCR35.6	PPCR35.5	PPCR35.4	PPCR35.3	PPCR35.2	PPCR35.1	PPCR35.0	1111 1111
PPCR36	0F24h	PPCR36.7	PPCR36.6	PPCR36.5	PPCR36.4	PPCR36.3	PPCR36.2	PPCR36.1	PPCR36.0	0000 0000
PPCR37	0F25h	PPCR37.7	PPCR37.6	PPCR37.5	PPCR37.4	PPCR37.3	PPCR37.2	PPCR37.1	PPCR37.0	0000 0000
PPCR38	0F26h	PPCR38.7	PPCR38.6	PPCR38.5	PPCR38.4	PPCR38.3	PPCR38.2	PPCR38.1	PPCR38.0	0000 0000
PPCR39	0F27h	PPCR39.7	PPCR39.6	PPCR39.5	PPCR39.4	PPCR39.3	PPCR39.2	PPCR39.1	PPCR39.0	0000 0000
PPCR40	0F28h	PPCR40.7	PPCR40.6	PPCR40.5	PPCR40.4	PPCR40.3	PPCR40.2	PPCR40.1	PPCR40.0	0000 0000
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Note: PPCR Register bits can only be written, if the attached PPER register bit is low (resistors disabled).

MB91460 Series

30.4.7 Programmable Port Output Drive

The Ports listed in the following table have a programmable output drive option, which can be enabled bit-wise. The function is enabled by the Port Output Drive Registers (PODR).

Bit	Port Output Drive Registers	
	0 (default)	1
PODR _{x.y}	5 mA output drive	2 mA output drive

Table 30.4-11 PODR register list

	Addr	7	6	5	4	3	2	1	0	initial
PODR00	0E00h	PODR00.7	PODR00.6	PODR00.5	PODR00.4	PODR00.3	PODR00.2	PODR00.1	PODR00.0	0000 0000
PODR01	0E01h	PODR01.7	PODR01.6	PODR01.5	PODR01.4	PODR01.3	PODR01.2	PODR01.1	PODR01.0	0000 0000
PODR02	0E02h	PODR02.7	PODR02.6	PODR02.5	PODR02.4	PODR02.3	PODR02.2	PODR02.1	PODR02.0	0000 0000
PODR03	0E03h	PODR03.7	PODR03.6	PODR03.5	PODR03.4	PODR03.3	PODR03.2	PODR03.1	PODR03.0	0000 0000
PODR04	0E04h	PODR04.7	PODR04.6	PODR04.5	PODR04.4	PODR04.3	PODR04.2	PODR04.1	PODR04.0	0000 0000
PODR05	0E05h	PODR05.7	PODR05.6	PODR05.5	PODR05.4	PODR05.3	PODR05.2	PODR05.1	PODR05.0	0000 0000
PODR06	0E06h	PODR06.7	PODR06.6	PODR06.5	PODR06.4	PODR06.3	PODR06.2	PODR06.1	PODR06.0	0000 0000
PODR07	0E07h	PODR07.7	PODR07.6	PODR07.5	PODR07.4	PODR07.3	PODR07.2	PODR07.1	PODR07.0	0000 0000
PODR08	0E08h	PODR08.7	PODR08.6	PODR08.5	PODR08.4	PODR08.3	PODR08.2	PODR08.1	PODR08.0	0000 0000
PODR09	0E09h	PODR09.7	PODR09.6	PODR09.5	PODR09.4	PODR09.3	PODR09.2	PODR09.1	PODR09.0	0000 0000
PODR10	0E0Ah	PODR10.7	PODR10.6	PODR10.5	PODR10.4	PODR10.3	PODR10.2	PODR10.1	PODR10.0	0000 0000
PODR11	0E0Bh	PODR11.7	PODR11.6	PODR11.5	PODR11.4	PODR11.3	PODR11.2	PODR11.1	PODR11.0	0000 0000
PODR12	0E0Ch	PODR12.7	PODR12.6	PODR12.5	PODR12.4	PODR12.3	PODR12.2	PODR12.1	PODR12.0	0000 0000
PODR13	0E0Dh	PODR13.7	PODR13.6	PODR13.5	PODR13.4	PODR13.3	PODR13.2	PODR13.1	PODR13.0	0000 0000
PODR14	0E0Eh	PODR14.7	PODR14.6	PODR14.5	PODR14.4	PODR14.3	PODR14.2	PODR14.1	PODR14.0	0000 0000
PODR15	0E0Fh	PODR15.7	PODR15.6	PODR15.5	PODR15.4	PODR15.3	PODR15.2	PODR15.1	PODR15.0	0000 0000
PODR16	0E10h	PODR16.7	PODR16.6	PODR16.5	PODR16.4	PODR16.3	PODR16.2	PODR16.1	PODR16.0	0000 0000
PODR17	0E11h	PODR17.7	PODR17.6	PODR17.5	PODR17.4	PODR17.3	PODR17.2	PODR17.1	PODR17.0	0000 0000
PODR18	0E12h	PODR18.7	PODR18.6	PODR18.5	PODR18.4	PODR18.3	PODR18.2	PODR18.1	PODR18.0	0000 0000
PODR19	0E13h	PODR19.7	PODR19.6	PODR19.5	PODR19.4	PODR19.3	PODR19.2	PODR19.1	PODR19.0	0000 0000
PODR20	0E14h	PODR20.7	PODR20.6	PODR20.5	PODR20.4	PODR20.3	PODR20.2	PODR20.1	PODR20.0	0000 0000
PODR21	0E15h	PODR21.7	PODR21.6	PODR21.5	PODR21.4	PODR21.3	PODR21.2	PODR21.1	PODR21.0	0000 0000
PODR22	0E16h	PODR22.7	PODR22.6	PODR22.5	PODR22.4	PODR22.3	PODR22.2	PODR22.1	PODR22.0	0000 0000
PODR23	0E17h	PODR23.7	PODR23.6	PODR23.5	PODR23.4	PODR23.3	PODR23.2	PODR23.1	PODR23.0	0000 0000
PODR24	0E18h	PODR24.7	PODR24.6	PODR24.5	PODR24.4	PODR24.3	PODR24.2	PODR24.1	PODR24.0	0000 0000
PODR25	0E19h	PODR25.7	PODR25.6	PODR25.5	PODR25.4	PODR25.3	PODR25.2	PODR25.1	PODR25.0	0000 0000

PODR26	0E1Ah	PODR26.7	PODR26.6	PODR26.5	PODR26.4	PODR26.3	PODR26.2	PODR26.1	PODR26.0	0000 0000
PODR27	0E1Bh	PODR27.7	PODR27.6	PODR27.5	PODR27.4	PODR27.3	PODR27.2	PODR27.1	PODR27.0	0000 0000
PODR28	0E1Ch	PODR28.7	PODR28.6	PODR28.5	PODR28.4	PODR28.3	PODR28.2	PODR28.1	PODR28.0	0000 0000
PODR29	0E1Dh	PODR29.7	PODR29.6	PODR29.5	PODR29.4	PODR29.3	PODR29.2	PODR29.1	PODR29.0	0000 0000
PODR30	0E1Eh	PODR30.7	PODR30.6	PODR30.5	PODR30.4	PODR30.3	PODR30.2	PODR30.1	PODR30.0	0000 0000
PODR31	0E1Fh	PODR31.7	PODR31.6	PODR31.5	PODR31.4	PODR31.3	PODR31.2	PODR31.1	PODR31.0	0000 0000
PODR32	0E20h	PODR32.7	PODR32.6	PODR32.5	PODR32.4	PODR32.3	PODR32.2	PODR32.1	PODR32.0	0000 0000
PODR33	0E21h	PODR33.7	PODR33.6	PODR33.5	PODR33.4	PODR33.3	PODR33.2	PODR33.1	PODR33.0	0000 0000
PODR34	0E22h	PODR34.7	PODR34.6	PODR34.5	PODR34.4	PODR34.3	PODR34.2	PODR34.1	PODR34.0	0000 0000
PODR35	0E23h	PODR35.7	PODR35.6	PODR35.5	PODR35.4	PODR35.3	PODR35.2	PODR35.1	PODR35.0	0000 0000
PODR36	-	-	-	-	-	-	-	-	-	---- ----
PODR37	0E25h	PODR37.7	PODR37.6	PODR37.5	PODR37.4	PODR37.3	PODR37.2	PODR37.1	PODR37.0	0000 0000
PODR38	-	-	-	-	-	-	-	-	-	---- ----
PODR39	-	-	-	-	-	-	-	-	-	---- ----
PODR40	-	-	-	-	-	-	-	-	-	---- ----
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

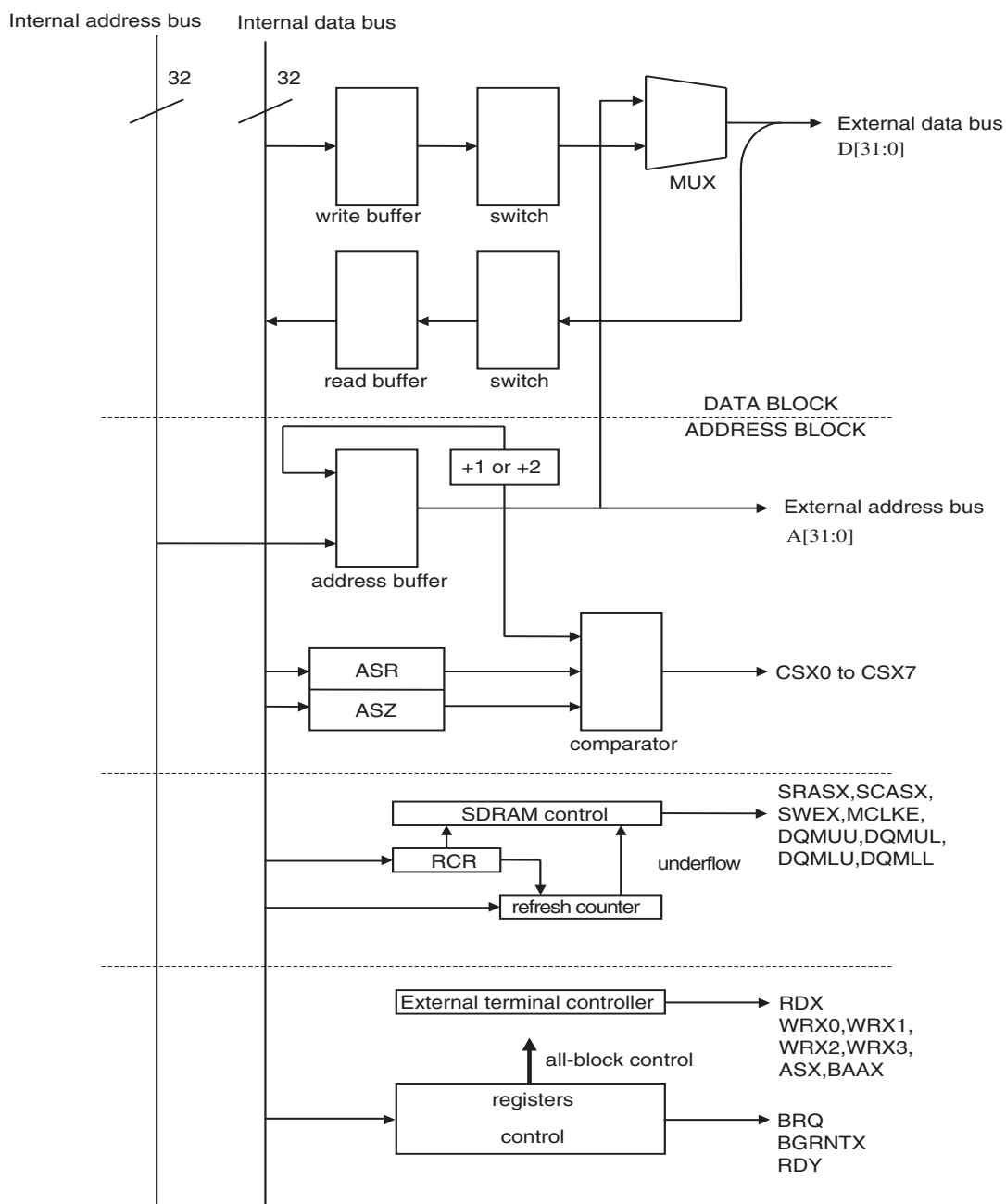
Chapter 31 External Bus

The external bus interface controller controls the interfaces between the internal bus and the external memory and I/O devices.

This chapter explains each function of the external bus interface and its operation.

31.1. Overview of the External Bus Interface

Figure 31.1-1 Block diagram of External bus interface



31.1.1 Features

- The external bus interface has the following features:
 - Addresses of up to 32 bits (4 GB space) can be output.
 - Various kinds of external memory (8-bit/16-bit/32-bit modules) can be directly connected and multiple access timings can be mixed and controlled.
 - Asynchronous SRAM and asynchronous ROM/FLASH memory (multiple write strobe method or byte enable method)
 - Page mode ROM/FLASH memory (Page sizes 2, 4, and 8 can be used)
 - Burst mode ROM/FLASH memory (such as MBM29BL160D/161D/162D)
 - Address/data multiplex bus (8-bit/16-bit width only)
 - SDRAM (FCRAM modules are also supported, including two - and four - bank types with CAS latency 1 to 8)
 - Synchronous memory (such as ASIC built-in memory) (Synchronous SRAM cannot be directly connected)
 - Eight independent banks (chip select areas) can be set, and chip select corresponding to each bank can be output.
 - The size of each area can be set in multiples of 64 KB (64 KB to 2 GB for each chip select area).
 - An area can be set at any location in the logical address space (Boundaries may be limited depending on the size of the area.)
- In each chip select area, the following functions can be set independently:
 - Enabling and disabling of the chip select area (Disabled areas cannot be accessed)
 - Setting of the access timing type to support various kinds of memory
 - Detailed access timing setting (individual setting of the access type such as the wait cycle)
 - Setting of the data bus width (8-bit/16-bit)
 - Setting of the order of bytes (big or little endian) (Only big endian can be set for the CSX0 area)
 - Setting of write disable (read-only area)
 - Enabling and disabling of fetches from the built-in cache
 - Enabling and disabling of the prefetch function
 - Maximum burst length setting (1, 2, 4, 8)
- A different detailed timing can be set for each access timing type.
 - For the same type of access timing, a different setting can be made in each chip select area.
 - Auto-wait can be set to up to 15 cycles (asynchronous SRAM, ROM, Flash, and I/O area).
 - The bus cycle can be extended by external RDY input (asynchronous SRAM, ROM, Flash, and I/O area).
 - The first access wait and page wait can be set (burst, page mode, and ROM/FLASH area).
 - Various kinds of idle/recovery cycles and setting delays can be inserted.
 - Capable of setting timing values such as the CAS latency and RAS - CAS delay (SDRAM area)
 - Capable of controlling the distributed/centralized auto - refresh, self - refresh, and other refresh timings (SDRAM area)
- Fly-by transfer by DMA can be performed.
 - Transfer between memory and I/O can be performed in a single access operation.
 - The memory wait cycle can be synchronized with the I/O wait cycle in fly-by transfer.
 - The hold time can be secured by only extending transfer source access.
 - Idle/recovery cycles specific to fly-by transfer can be set.
- External bus arbitration using BRQ and BGRNTX can be performed.
- Pins that are not used by the external interface can be used as general-purpose I/O ports through settings.

MB91460 Series

31.1. Overview of the External Bus Interface

31.1.2 I/O Pins

The I/O pins are external bus interface pins (Some pins have other uses).

The following lists the I/O pins for each interface:

● Ordinary bus interface

- A31 to A00, D31 to D00 (AD15 to AD00)
- CSX0, CSX1, CSX2, CSX3, CSX4, CSX5, CSX6, CSX7
- ASX, SYSCLK, MCLKO
- RDX
- WEX, WRX0(UUBX), WRX1(ULBX), WRX2(LUBX), WRX3(LLBX)
- RDY, BRQ, BGRNTX

● Memory interface

- MCLKO, MCLKE
- MCLKI (for SDRAM)
- LBAX(=ASX), BAAX (for burst ROM/FLASH)
- SRASX, SCASX, SWEX (=WEX) (for SDRAM)
- DQMUU, DQMUL, DQMLU, DQMLL (for SDRAM (=WRX0, WRX1, WRX2, WRX3))

● DMA interface

- IOWRX, IORDX
- DACKX0, DACKX1, DACKX2, DACKX3
- DREQ0, DREQ1, DREQ2, DREQ3
- DEOP0, DEOP1, DEOP2, DEOP3

31.1.3 Register List

Figure 31.1-2 "List of External Bus Interface Registers" shows the registers used by the external bus interface:

Figure 31.1-2 List of External Bus Interface Registers

Address	31	24 23	16 15	08 07	00
00000640 _H	ASR0			ACR0	
00000644 _H	ASR1			ACR1	
00000648 _H	ASR2			ASR2	
0000064c _H	ASR3			ACR3	
00000650 _H	ASR4			ACR4	
00000654 _H	ASR5			ACR5	
00000658 _H	ASR6			ACR6	
0000065c _H	ASR7			ACR7	
00000660 _H	AWR0			AWR1	
00000664 _H	AWR2			AWR3	
00000668 _H	AWR4			AWR5	
0000066c _H	AWR6			AWR7	
00000670 _H	MCRA	MCRB		Reserved	Reserved
00000674 _H	Reserved	Reserved		Reserved	Reserved
00000678 _H	IOWR0	IOWR1		IOWR2	IOWR3
0000067c _H	Reserved	Reserved		Reserved	Reserved
00000680 _H	CSER	CHER		Reserved	TCR
00000684 _H	RCRH	RCRL		Reserved	Reserved
00000688 _H	Reserved	Reserved		Reserved	Reserved
0000068c _H	Reserved	Reserved		Reserved	Reserved
	Reserved	Reserved		Reserved	Reserved
000007f8 _H	Reserved	Reserved		Reserved	Reserved
000007fc _H	Reserved	(MODR)		Reserved	Reserved

*1: Reserved indicates a reserved register. Be sure to set "0".

*2: MODR cannot be accessed from user programs.

31.2. External Bus Interface Registers

This section explains the registers used in the external bus interface.

■ Register Types

The following registers are used by the external bus interface:

- Area select registers (ASR0-7)
- Area configuration registers (ACR0-7)
- Area wait registers (AWR0-7)
- Memory configuration register (MCRA for SDRAM/FCRAM auto - precharge OFF mode)
- Memory configuration register (MCRB for FCRAM auto - precharge ON mode)
- I/O wait registers for DMAC (IOWR0-3)
- Chip select enable register (CSER)
- Cache enable register (CHER)
- Refresh Control Register (RCR)
- Pin/timing control register (TCR)
- Mode register (MODR)

31.2.1 Area Select Registers 0-7(ASR0-7)

This section explains the configuration and functions of area select registers 0-7 (ASR0-7).

■ Configuration of area select registers 0-7 (ASR0-7)

The area select registers (ASR0-7: Area Select Registers 0-7) specify the start address of each chip select area of CSX0-CSX7.

[Figure 31.2-1 Configuration of the Area Select Registers \(ASR0-7\) \(Page No.660\)](#) shows the configuration of area select registers 0-7 (ASR0-7: Area Select Register).

Figure 31.2-1 Configuration of the Area Select Registers (ASR0-7)

ASR0	Initial value									Access
	15	14	13	12	...	2	1	0	INIT RST	
00000640 _H	A31	A30	A29	A18	A17	A16	0000 _H 0000 _H	W/R
ASR1	15	14	13	12	...	2	1	0		
00000644 _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR2	15	14	13	12	...	2	1	0		
00000648 _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR3	15	14	13	12	...	2	1	0		
0000064C _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR4	15	14	13	12	...	2	1	0		
00000650 _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR5	15	14	13	12	...	2	1	0		
00000654 _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR6	15	14	13	12	...	2	1	0		
00000658 _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R
ASR7	15	14	13	12	...	2	1	0		
0000065C _H	A31	A30	A29	A18	A17	A16	xxxx _H xxxx _H	W/R

■ Functions of Bits in the Area Select Registers (ASR0-7)

The start address can be set in the high-order 16 bits (bits A31-A16). Each chip select area starts with the address set in this register and covers the range set by the four bits ASZ3-0 of the ASR0-7 registers.

The boundary of each chip select area obeys the setting of the four bits ASZ3-0 of the ACR0-7 registers. For example, if an area of 1 MB is set by the four bits ASZ3-0, the low-order four bits of the ASR0-7 registers are ignored and only bits A31-20 are valid.

The ASR0 register is initialized to 0000_H by INIT and RST. ASR1-7 are not initialized by INIT and RST, and are therefore undefined. After starting chip operation, be sure to set the corresponding ASR register before enabling each chip select area with the CSER register.

MB91460 Series

31.2.2 Area Configuration Registers 0-7 (ACR0-7)

This section explains the configuration and functions of area configuration registers 0-7 (ACR0-7).

■ Configuration of Area Configuration Registers 0-7 (ACR0-7)

The area configuration registers 0-7 (ACR0-7: Area Configuration Register 0-7) set the function of each chip select area.

Figure 31.2-2 “Configuration of Area Configuration Registers 0-7 (ACR0-7)” shows the configuration of area configuration registers 0-7 (ACR0-7).

Figure 31.2-2 Configuration of Area Configuration Registers 0-7 (ACR0-7) (Continued on next page)

									Initial value		
ACR0H	15	14	13	12	11	10	9	8	INIT	RST	Access
00000642 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	1111**00 _B	1111**00 _B	W/R
ACR0L	7	6	5	4	3	2	1	0			
00000643 _H	SREN	PFEN	WREN	0	TYP3	TYP2	TYP1	TYP0	00000000 _B	00000000 _B	W/R
ACR1H	15	14	13	12	11	10	9	8			
00000646 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	Xxxxxxxx _B	xxxxxxx _B	W/R
ACR1L	7	6	5	4	3	2	1	0			
00000647 _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	Xxxxxxxx _B	W/R
ACR2H	15	14	13	12	11	10	9	8			
0000064A _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR2L	7	6	5	4	3	2	1	0			
0000064B _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	xxxxxxx _B	W/R
ACR3H	15	14	13	12	11	10	9	8			
0000064E _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR3L	7	6	5	4	3	2	1	0			
0000064F _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	Xxxxxxxx _B	xxxxxxx _B	W/R

								Initial value		Access	
								INIT	RST		
ACR4H	15	14	13	12	11	10	9	8			
00000652 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR4L	7	6	5	4	3	2	1	0			
00000653 _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	xxxxxxx _B	W/R
ACR5H	15	14	13	12	11	10	9	8			
00000656 _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR5L	7	6	5	4	3	2	1	0			
00000657 _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	xxxxxxx _B	W/R
ACR6H	15	14	13	12	11	10	9	8			
0000065A _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR6L	7	6	5	4	3	2	1	0			
0000065B _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	xxxxxxx _B	W/R
ACR7H	15	14	13	12	11	10	9	8			
0000065E _H	ASZ3	ASZ2	ASZ1	ASZ0	DBW1	DBW0	BST1	BST0	xxxxxxx _B	xxxxxxx _B	W/R
ACR7L	7	6	5	4	3	2	1	0			
0000065F _H	SREN	PFEN	WREN	LEND	TYP3	TYP2	TYP1	TYP0	xxxxxxx _B	xxxxxxx _B	W/R

The following explains the function of each bit:

[Bits 15-12] ASZ3-0 (Area Size Bits 3-0)

These bits set the area size. [Table 31.2-1](#) "Area Size Settings" shows their settings.

Table 31.2-1 Area Size Settings

ASZ3	ASZ2	ASZ1	ASZ0	Size of each chip select area
0	0	0	0	64 KB (00010000 _H byte, ASR A[31:16] bits are valid)
0	0	0	1	128 KB (00020000 _H byte, ASR A[31:17] bits are valid)
0	0	1	0	256 KB (00040000 _H byte, ASR A[31:18] bits are valid)
0	0	1	1	512 KB (00080000 _H byte, ASR A[31:19] bits are valid)
0	1	0	0	1 MB (00100000 _H byte, ASR A[31:20] bits are valid)
0	1	0	1	2 MB (00200000 _H byte, ASR A[31:21] bits are valid)
0	1	1	0	4 MB (00400000 _H byte, ASR A[31:22] bits are valid)
0	1	1	1	8 MB (00800000 _H byte, ASR A[31:23] bits are valid)

Table 31.2-1 Area Size Settings

ASZ3	ASZ2	ASZ1	ASZ0	Size of each chip select area
1	0	0	0	16 MB (01000000 _H byte, ASR A[31:24] bits are valid)
1	0	0	1	32 MB (02000000 _H byte, ASR A[31:25] bits are valid)
1	0	1	0	64 MB (04000000 _H byte, ASR A[31:26] bits are valid)
1	0	1	1	128 MB (08000000 _H byte, ASR A[31:27] bits are valid)
1	1	0	0	256 MB (10000000 _H byte, ASR A[31:28] bits are valid)
1	1	0	1	512 MB (20000000 _H byte, ASR A[31:29] bits are valid)
1	1	1	0	1024 MB (40000000 _H byte, ASR A[31:30] bits are valid)
1	1	1	1	2048 MB (80000000 _H byte, ASR A[31] bit is valid)

ASZ3-0 are used to set the size of each area by modifying the number of bits for address comparison to a value different from ASR. Thus, an ASR contains bits that are not compared. Bits ASZ3-0 of ACR0 are initialized to 1111_B (0F_H) by RST. Despite this setting, however, the CSX0 area just after RST is executed is specially set from 00000000_H to FFFFFFFF_H (setting of entire area). The entire-area setting is reset after the first write to ACR0 and an appropriate size is set as indicated in Table 31.2-1 "Area Size Settings".

[Bits 11-10] DBW1-0 (Data Bus Width 1-0)

These bits set the data bus width of each chip select area as indicated in Table 31.2-2 "Setting of the Data Bus Width of Each Chip Select Area":

Table 31.2-2 Setting of the Data Bus Width of Each Chip Select Area

DBW1	DBW0	Data bus width
0	0	8 bits (byte access)
0	1	16 bits (halfword access)
1	0	32 bits (word access)
1	1	Reserved Setting disabled

The same values as those of the WTH bits of the mode vector are written automatically to bits DBW1-0 of ACR0 during the reset sequence.

[Bits 9-8] BST1-0 (Burst Size 1-0)

These bits set the maximum burst length of each chip select area as indicated in Table 31.2-3 "Setting of the Maximum Burst Length of Each Chip Select".

Table 31.2-3 Setting of the Maximum Burst Length of Each Chip Select

BST1	BST0	Maximum burst length
0	0	1 (single access)
0	1	2 bursts (address boundary: 1 bit)
1	0	4 bursts (address boundary: 2 bits)
1	1	8 bursts (address boundary: 3 bits)

In areas for which a burst length other than the single access is set, continuous burst access is performed within the address boundary determined by the burst length only when prefetch access is performed or data having a size exceeding the bus width is read.

Setting of 2 bursts or less as the maximum burst length in the bus width 16-bit area is recommended.

RDY input is ignored in areas for which any burst length other than the single access is set.

[Bit 7] SREN (ShaRed Enable)

This bit sets enabling or disabling of sharing of each chip select area by BRQ/BGRNTX as indicated in the following table.

SREN	Sharing enable/disable
0	Disable sharing by BRQ/BGRNTX (CSXn cannot be high impedance)
1	Enable sharing by BRQ/BGRNTX (CSXn can be high impedance)

In areas where sharing is enabled, chip select output (CSXn) is set to high impedance while the bus is open (during BGRNTX=Low output). In areas where sharing is disabled, chip select output (CSXn) is not set to high impedance even though the bus is open (during BGRNTX=Low output).

Access strobe output (ASX, BAAX, RDX, WRX0, WRX1, WRX2, WRX3, WEX, MCLKX, MCLKEX) is set to high impedance only if sharing of all areas enabled by CSER is enabled.

[Bit 6] PFEN (PreFetch Enable)

This bit sets enabling and disabling of prefetching of each chip select area as indicated in the following table.

PFEN	Prefetch enable/disable
0	Disable prefetch
1	Enable prefetch

When reading from an area for which prefetching is enabled, the subsequent address is read in advance and stored in the built-in prefetch buffer. When the stored address is accessed from the internal bus, the lookahead data in the prefetch buffer is returned without performing external access.

For more information, see Section 31.8."Prefetch Operation".

[Bit 5] WREN (WRite Enable)

This bit sets enabling and disabling of writing to each chip select area.

WREN	Write enable/disable
0	Disable write
1	Enable write

If an area for which write operations are disabled is accessed for a write operation from the internal bus, the access is ignored and no external access at all is performed. Set the WREN bit of areas for which write operations are not required, such as data areas, to 0.

MB91460 Series**[Bit 4] LEND (Little ENDian select)**

This bit sets the order of bytes of each chip select area as indicated in the following table.

LEND	Order of bytes
0	Big endian
1	Little endian

Be sure to set the LEND bit of ACR0 to 0. CSX0 supports only the big endian method.

[Bits 3-0] TYP3-0 (TYPE select)

These bits set the access type of each chip select area as indicated in [Table 31.2-4 "Access Type Settings for Each Chip Select Area"](#).

Table 31.2-4 Access Type Settings for Each Chip Select Area

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM, I/O, and single/page/burst-ROM/FLASH)
0	1	x	x	Address data multiplex access (8/16-bit bus width only)
0	x	x	0	Disable WAIT insertion by the RDY pin.
0	x	x	1	Enable WAIT insertion by the RDY pin (disabled during bursts).
0	x	0	x	Use the WRX0-WRX3 pins as write strobes (WEX is always H).
0	x	1	x	Use the WEX pin as the write strobe. ^{*1}
1	0	0	0	Memory type A: SDRAM/FCRAM
1	0	0	1	Memory type B: FCRAM
1	0	1	0	Setting disabled
1	0	1	1	Setting disabled
1	1	0	0	Setting disabled
1	1	0	1	Setting disabled
1	1	1	0	Setting disabled
1	1	1	1	Mask area setting (The access type is the same as that of the overlapping area) ^{*3}

^{*1}: If this setting is made, WRX0-WRX3 can be used as the enable of each bit.

^{*3}: See the CS area mask setting function (next bullet).

Set the access type as the combination of all bits.

For details of the operations of each access type, see the explanation of operation of each type.

● CS area mask setting function

If you want to set an area some of whose operation settings are changed for a certain CS area (referred to as the base setting area), you can set TYPE3-0 of ARC in another CS area to 1111 so that the area can function as a mask setting area.

If the mask setting function is not used, disable any overlapping area settings for multiple CS areas.

Access operations to the mask setting area are as follows:

- CSX corresponding to a mask setting area is not asserted.
- CSX corresponding to a base setting area is not asserted.
- For the following ACR settings, the settings on the mask setting area side are valid:
 - Bits 11-10 (DBW1-0): Bus width setting
 - Bits 9-8 (BST1-0): Burst length setting
 - Bit 7 (SREN): Sharing-enable setting
 - Bit 6 (PFEN): Prefetch-enable setting
 - Bit 5 (WREN): Write-enable setting (For this setting only, only a setting that is the same as that of the base setting area is allowed)
 - Bit 4 (LEND): Little endian setting
- For the following ACR setting, the setting on the base setting area side is valid:
 - Bits 3-0 (TYPE3-0): Access type setting
- For the AWR settings, the settings on the mask setting area side are valid.
- For the CHER settings, the settings on the mask setting area side are valid.

A mask setting area can be set for only part of another CS area (base setting area). A mask setting area cannot be set for an area without a base setting area. Take care when setting ASR and bits ASZ3-0 of ACR.

The following restrictions apply when using these bits:

- A write-enable setting cannot be implemented by a mask.
- Write-enable settings in the base CS area and the mask setting area must be identical.
- If write operations to a mask setting area are disabled, the area is not masked and operates as a base CS area.
- If write operations to the base CS area are disabled but are enabled to the mask setting area, the area has no base, resulting in malfunctions.

MB91460 Series

31.2.3 Area Wait Register (AWR0-7)

This section explains the configuration and functions of the area wait registers (AWR0-7).

■ Configuration of the Area Wait Registers (AWR0-7)

The area wait registers (AWR0-7: Area Wait Register 0-7) specify various kinds of waits for each chip select area.

Figure 31.2-3 "Configuration of the Area Wait Registers (AWR0-7)" shows the configuration of the area wait registers (AWR0-7).

Figure 31.2-3 Configuration of the Area Wait Registers (AWR0-7) (Continued on next page)

								Initial value		Access	
								INIT	RST		
AWR0H	31	30	29	28	27	26	25	24			
00000660 _H	W15	W14	W13	W12	W11	W10	W09	W08	01111111 _b	01111111 _b	W/R
AWR0L	23	22	21	20	19	18	17	16			
00000661 _H	W07	W06	W05	W04	W03	W02	W01	W00	11111011 _B	11111011 _B	W/R
AWR1H	15	14	13	12	11	10	9	8			
00000662 _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxx _b	xxxxxxx _b	W/R
AWR1L	7	6	5	4	3	2	1	0			
00000663 _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxx _b	xxxxxxx _b	W/R
AWR2H	31	30	29	28	27	26	25	24			
00000664 _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxx _b	xxxxxxx _b	W/R
AWR2L	23	22	21	20	19	18	17	16			
00000665 _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxx _b	xxxxxxx _b	W/R
AWR3H	15	14	13	12	11	10	9	8			
00000666 _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxx _b	xxxxxxx _b	W/R
AWR3L	7	6	5	4	3	2	1	0			
00000667 _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxx _b	xxxxxxx _b	W/R
AWR4H	31	30	29	28	27	26	25	24			
00000668 _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxx _b	xxxxxxx _b	W/R
AWR4L	23	22	21	20	19	18	17	16			
00000669 _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxx _b	xxxxxxx _b	W/R

								Initial value			
								INIT	RST	Access	
AWR5H	15	14	13	12	11	10	9	8			
0000066A _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxxx _b	xxxxxxxx _b	W/R
AWR5L	7	6	5	4	3	2	1	0			
0000066B _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxxx _b	xxxxxxxx _b	W/R
AWR6H	31	30	29	28	27	26	25	24			
0000066C _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxxx _b	xxxxxxxx _b	W/R
AWR6L	23	22	21	20	19	18	17	16			
0000066D _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxxx _b	xxxxxxxx _b	W/R
AWR7H	15	14	13	12	11	10	9	8			
0000066E _H	W15	W14	W13	W12	W11	W10	W09	W08	xxxxxxxx _b	xxxxxxxx _b	W/R
AWR7L	7	6	5	4	3	2	1	0			
0000066F _H	W07	W06	W05	W04	W03	W02	W01	W00	xxxxxxxx _b	xxxxxxxx _b	W/R

The function of each bit changes according to the access type (TYP(3-0) bits) setting of the ACR0-7 registers. A chip select area determined by either of the following settings becomes the area for normal access or a address/data multiplex access operation.

TYP3	TYP2	TYP1	TYP0	Access type
0	0	x	x	Normal access (asynchronous SRAM, I/O, and single/page/burst-ROM/FLASH)
0	1	x	x	Address data multiplex access (8/16-bit bus width only)

The following lists the functions of each AWR0-7 bit for a normal access or address/data multiplex access area. Since the initial values of registers other than AWR0 are undefined, set them to their initial values before enabling each area with the CSER register.

The following explains the functions of the bits in the area wait registers (AWR0-7).

[Bits 15-12] W15-12 (First Wait Cycle)

These bits set the number of auto-wait cycles to be inserted into the first access cycle of each cycle. Except for the burst access cycles, only this wait setting is used.

MB91460 Series

[Table 31.2-5](#) "Settings for the Number of Auto-Wait Cycles (During First Access)" lists the settings for the number of auto-wait cycles during first access.

Table 31.2-5 Settings for the Number of Auto-Wait Cycles (During First Access)

W15	W14	W13	W12	First access wait cycle
0	0	0	0	Auto-wait cycle 0
0	0	0	1	Auto-wait cycle 1
...				...
1	1	1	1	Auto-wait cycle 15

[Bits 11-8] W11-08 (Inpage Access Wait Cycle)

These bits set the number of auto-wait cycles to be inserted into the inpage access cycle during burst access. They are valid only for burst cycles.

[Table 31.2-6](#) "Settings for the Number of Auto-Wait Cycles (During Burst Access)" lists the settings for the number of auto-wait cycles during burst access.

Table 31.2-6 Settings for the Number of Auto-Wait Cycles (During Burst Access)

W11	W10	W09	W08	Inpage access wait cycle
0	0	0	0	Auto-wait cycle 0
0	0	0	1	Auto-wait cycle 1
...				...
1	1	1	1	Auto-wait cycle 15

If the same value is set for the first access wait cycle and inpage access wait cycle, the access time for the address in each access cycle is not the same. This is because the inpage access cycle contains an address output delay.

[Bits 7,6] W07-06 (Read -> Write Idle Cycle)

The read -> write idle cycle is set to prevent collision of read data and write data on the data bus when a write cycle follows a read cycle. During an idle cycle, all chip select signals are negated and the data terminals maintain the high impedance state. If a write cycle follows a read cycle or an access operation to another chip select area occurs after a read cycle, the specified idle cycle is inserted. [Table 31.2-7](#) "Settings of the Idle Cycle" lists the settings for idle cycles.

Table 31.2-7 Settings of the Idle Cycle

W07	W06	Read -> write idle cycles
0	0	0 cycle
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

[Bits 5, 4] W05, W04 (Write Recovery Cycle)

The write recovery cycle is set if a device that limits the access period after write access is to be controlled. During a write recovery cycle, all chip select signals are negated and the data pins maintain the high impedance state. If the write recovery cycle is set to 1 or more, a write recovery cycle is always inserted after write access.

Table 31.2-8 "Settings for the Number of Write Recovery Cycles" lists the settings for the number of write recovery cycles.

Table 31.2-8 Settings for the Number of Write Recovery Cycles

W05	W04	Write recovery cycles
0	0	0 cycle
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

[Bit 3] W03 (WRX0-WRX3, WEX Output Timing Selection)

The WRX0-WRX3, WEX output timing setting selects whether to use write strobe output as an asynchronous strobe or synchronous write enable. The asynchronous strobe setting corresponds to normal memory/IO. The synchronous enable setting corresponds to clock-synchronized memory/IO (such as the memory in an ASIC).

W03	WRX0-WRX3, WEX output timing selection
0	MCLKO synchronous write enable output (valid from ASX=L)
1	Asynchronous write strobe output (normal operation)

If synchronous write enable (W03 bit of AWR is 0) is used, operations are as follows:

- The timing of synchronous write enable output assumes that the output is captured by the rising edge of MCLKO output of an external memory access clock. This timing is different from the asynchronous strobe output timing.
- The WRX0-WRX3 and WEX terminal output asserts synchronous write enable output at the timing at which ASX pin output is asserted. For a write to an external bus, the synchronous write enable output is L. For a read from an external bus, the synchronous write enable output is H.
- Write data is output from the external data output pin in the clock cycle following the cycle in which synchronous write enable output is asserted. If write data cannot be output because the internal bus is temporarily unavailable, assertion of synchronous write enable output may be extended until write data can be output.
- Read strobe output (RDX) functions as an asynchronous read strobe regardless of the setting of the WRX0-WRX3 and WEX output timing. Use it as is for controlling the data I/O direction.

If synchronous write enable output is used, the following restrictions apply:

- Do not make the following additional wait settings:
 - CSXn -> RDX/WRXn/WEX setup (Always set 0 for the W01 bit of AWR)
 - First wait cycle setting (Always set 0000_B for the W15-W12 bits of AWR)
- Do not make the following access type settings (TYPE3-0 bits in the ACR register (bits 3-0))

- Address/data multiplex bus setting (Always set 0 for the TYPE2 bit of ACR)
- Setting to use WRX0-WRX3 as a strobe (Always set 0 for the TYPE1 bit of ACR)
- RDY input enable setting (Always set 0 for the TYPE0 bit of ACR)
- For synchronous write enable output, always set 1(00_B for bits BST1-0 bits of ACR) as the burst length.

[Bit 2] W02 (Address -> CSXn Delay)

The address -> CSXn delay setting is made when a certain type of setup is required for the address when CSXn falls or CSXn edges are needed for successive accesses to the same chip select area.

Set the address and set the delay from ASX output to CSX0-CSX7 output.

W02	Address -> CSXn delay
0	No delay
1	Delay

If no delay is selected by setting 0, assertion of CSX0-CSX7 starts at the same timing that ASX is asserted. If, at this point, successive accesses are made to the same chip select area, assertion of CSX0-CSX7 without change between two access operations may continue.

If delay is specified by selecting 1, assertion of CSX0-CSX7 starts when the external clock memory MCLKO output rises. If, at this point, successive accesses are made to the same chip select area, CSX0-CSX7 are negated at a timing between two access operations. If CSX delay is selected, one setup cycle is inserted before asserting the read/write strobe after assertion of the delayed CSXn (operation is the same as the CSXn ->RDX/WEX setup setting of W01).

The address -> CSXn delay setting works for DACKX signal (basic mode) output to the same area in the same way. DACKX output in basic mode has the same waveforms as those of CSX output to the same area.

[Bit 1] W01 (CSXn -> RDX/WRXn/WEX Setup Extension Cycle)

The CSXn -> RDX/WRXn/WEX setup extension cycle is set to extend the period before the read/write strobe is asserted after CSXn is asserted. At least one setup extension cycle is inserted before the read/write strobe is asserted after CSX is asserted.

W01	CSXn -> RDX/WRXn/WEX setup delay cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting 0, RDX/WRX0-WRX3/WEX are output at the earliest when external clock MCLKO output rises just after CSX is asserted. WRX0-WRX3/WEX may be delayed one cycle or more depending on the internal bus state.

If 1 cycle is selected by setting 1, RDX/WRX0-WRX3/WEX are always output 1 cycle or more later.

When successive accesses are made within the same chip select area without negating CSXn, a setup extension cycle is not inserted. If a setup extension cycle for determining the address is required, set the W02 bit and insert the address -> CSXn delay. Since CSXn is negated for each access operation, the setup extension cycle is enabled.

If the CSXn delay set by W02 is inserted, this setup cycle is always enabled regardless of the setting of the W01 bit.

[Bit 0] W00 (RDX/WRXn/WEX -> CSXn Hold Extension Cycle)

The RDX/WRXn/WEX -> CSXn hold extension cycle is set to extend the period before negating CSXn after the read/write strobe is negated. One hold extension cycle is inserted before CSXn is negated after the read/write strobe is negated.

W00	RDX/WRXn/WEX -> CSXn hold extension cycle
0	0 cycle
1	1 cycle

If 0 cycle is selected by setting 0, CSX0-CSX7 are negated after the hold delay after it starts on the rising edge of external memory clock MCLKO output after RDX/WRX0-WRX3/WEX are negated.

If 1 cycle is selected by setting 1, CSX0-CSX7 are negated one cycle later.

When making successive accesses within the same chip select area without negating CSXn, the hold extension cycle is not inserted. If a hold extension cycle for determining the address is required, set the W02 bit and insert the address -> CSXn delay. Since CSXn is negated for each access operation, this hold extension cycle is enabled.

● **Memory type A(SDRAM/FCRAM) and Memory type B(FCRAM)**

The chip select areas for which the access type (TYP3 to TYP0 bits) in the ACRn registers has been set as in [Table 31.2-9](#) serve for SDRAM/FCRAM access.

[Table 31.2-9](#) lists the access type settings (TYP3 to TYP0 bits).

Table 31.2-9 Access Type Settings (TYP3 - TYP0 Bits)

TYP3	TYP2	TYP1	TYP0	Access type
1	0	0	0	Memory type A: SDRAM/FCRAM (Auto - precharge is not used.)

The following explains those functions of individual bits in AWRn which apply to SDRAM access areas. As the initial value is undefined, set the access type before each area is enabled by the chip select area enable register (CSER).

For all the areas connected to SDRAM/FCRAM, use the same settings for this type of registers.

The following summarizes the functions of individual bits in the area wait registers AWRn.

[Bit 15] W15: Reserved bit

Be sure to set this bit to 0.

[Bits 14 - 12] W14 to W12 (RAS - CAS delay Cycle): RAS - CAS delay cycles

Set these bits to the number of cycles from RAS output to CAS output.

[Table 31.2-10](#) lists the settings for the number of cycles from RAS output to CAS output.

Table 31.2-10 Setting the Number of Cycles from RAS Output to CAS Output

W14	W13	W12	RAS-CAS delay cycle
0	0	0	1 cycle
0	0	0	2 cycles
...			...
1	1	1	8 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same RAS - CAS delay cycle.

MB91460 Series**[Bit 11] W11: Reserved bit**

Be sure to set this bit to 0.

[Bits 10 - 8] W10 to W08 (CAS latency Cycle): CAS latency

Set these bits to the CAS latency.

[Table 31.2-11](#) lists the settings for the CAS latency.

Table 31.2-11 CAS Latency Setting

W10	W09	W08	CAS latency
0	0	0	1 cycle
0	0	0	2 cycles
...			...
1	1	1	8 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same CAS latency.

[Bits 7 - 6] W07 and W06 (Read - >Write Cycle): Read - to - write cycle

Set these bits to the minimum number of cycles from the last read data input cycle to the write command issuance. Set the minimum number of cycles taken until issuance.

[Table 31.2-12](#) lists the settings for the read - to - write cycle.

Table 31.2-12 Read - to - write cycle

W07	W06	Read - to - write cycle
0	0	1 cycle
0	1	2 cycles
1	0	3 cycles
1	1	4 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same read - to - write cycle.

The number of read - to - write idle cycles is one smaller than the number of cycles set by this bit.

[Bits 5 - 4] W05 and W04 (Write Recovery Cycle): Write recovery cycle

Set these bits to the minimum number of cycles from the last write data output to the next read command issuance.

[Table 31.2-13](#) lists the settings for the write recovery cycle.

Table 31.2-13 Write recovery cycle

W05	W04	Write recovery cycle
0	0	Prohibited
0	1	2 cycles
1	0	3 cycles

Table 31.2-13 Write recovery cycle

W05	W04	Write recovery cycle
1	1	4 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same write recovery cycle.

[Bits 3 - 2] W03 and W02 (RAS Active time): RAS active time

Set these bits to the minimum number of cycles for RAS active time.

Table 31.2-14 lists the settings for RAS active time.

Table 31.2-14 RAS active time

W03	W02	RAS active time
0	0	1 cycle
0	1	2 cycles
1	0	5 cycles
1	1	6 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same RAS active time.

[Bits 1 - 0] W01 and W00 (RAS precharge cycle): RAS precharge cycles

Set these bits to the number of RAS precharge cycles.

Table 31.2-15 lists the settings for the RAS precharge cycle.

Table 31.2-15 RAS precharge cycle

W03	W02	RAS precharge cycle
0	0	1 cycle
0	1	2 cycles
1	0	3 cycles
1	1	4 cycles

For all the areas connected to SDRAM/FCRAM, set these bits to the same RAS precharge cycle.

31.2.4 Memory setting register (MCRA for SDRAM/FCRAM auto - precharge OFF mode)

This section describes the configuration and the function of memory setting register (MCRA for SDRAM/FCRAM auto - precharge OFF mode).

■ Structure of the Memory Setting Register (MCRA for SDRAM/FCRAM auto - precharge OFF mode)

Memory setting register (MCRA for SDRAM/FCRAM auto - precharge OFF mode)

The memory setting register (MCRA: Memory Setting Register for extend type - A for SDRAM/FCRAM auto - precharge OFF mode) is used to make various settings for SDRAM/FCRAM connected to the chip select area.

Figure 31.2-4 shows the bit configuration of the memory setting register (MCRA for SDRAM/FCRAM auto - precharge OFF mode).

Figure 31.2-4 Bit configuration of the MCRA register (for SDRAM/FCRAM auto - precharge OFF mode)

bit	31	30	29	28	27	26	25	24	Initial value
Address 00000670H	Reserved	PSZ2	PSZ1	PSZ0	WBST	BANK	ABS1	ABS0	XXXXXXXX _B (INIT) XXXXXXXX _B (RST)
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

The register serves as the area for making various settings for SDRAM/FCRAM connected to the chip select area for which the access type (TYP3 to TYP0 bits) in the ACRn registers has been set as in Table 31.2-25 .

Table 31.2-25 lists the access type settings (TYP3 to TYP0 bits).

Table 31.2-16 Access type settings (TYP3 to TYP0 bits)

TYP3	TYP2	TYP1	TYP0	Access type
1	0	0	0	Memory type A:SDRAM/FCRAM (not used auto precharge)

MCRB shares register hardware with MCRA. Updating the MCRA therefore updates the MCRB accordingly.

The following summarizes the functions of individual bits in the memory setting register (MCRA for SDRAM/FCRAM auto - precharge OFF mode).

[Bit 31] Reserved bit

Be sure to set this bit to 0.

[Bits 30 - 28] PSZ2, PSZ1, PSZ0 (Page SiZe): Page size

Set these bits to the page size of SDRAM to be connected.

Table 31.2-26 lists the settings for the page size of SDRAM connected.

Table 31.2-17 Settings for the page size of SDRAM

PSZ2	PSZ1	PSZ0	Page size of SDRAM
0	0	0	8-bit column address:A0 to A7(256 memory words)
0	0	1	9-bit column address:A0 to A8(512 memory words)
0	1	0	10-bit column address:A0 to A9(1024 memory words)
0	1	1	11-bit column address:A0 to A9, A11(2048memory words)
1	X	X	Prohibited

[Bit 27] WBST (Write BurST enable): Write burst setting

Set this bit to select whether to burst - write for write access.

Table 31.2-27 lists the settings for burst write.

Table 31.2-18 Settings for burst write

WBST	Settings for burst write
0	Single write
1	Burst write

For connecting FCRAM, be sure to set the bit to 1.
FCRAM supports neither burst read nor single write mode.

[Bit 26] BANK (BANK type select): Bank number setting

Set this bit to the number of banks of SDRAM to be connected.

Table 31.2-28 lists the settings for bank number.

Table 31.2-19 settings for bank number

BANK	Settings for bank number
0	2 banks
1	4 banks

[Bits 25 - 24] ABS1, ABS0 (Active Bank Select): Setting of active bank number

Set these bits to the maximum number of banks to be made active simultaneously.

Table 31.2-29 lists the settings for the number of active banks.

Table 31.2-20 Settings for the number of active banks

ABS1	ABS0	Number of active banks
0	0	1 bank
0	1	2 banks
1	0	3 banks
1	1	4 banks

31.2.5 Memory setting register (MCRB for FCRAM auto - precharge ON mode)

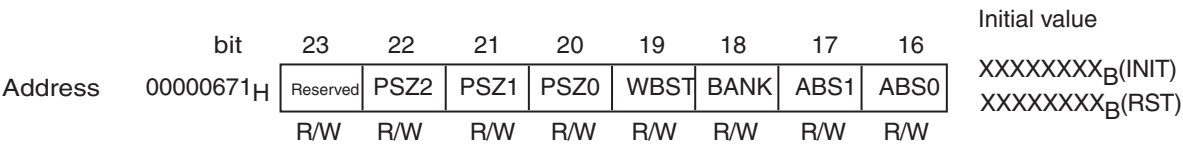
This section describes the memory setting register (MCRB for FCRAM auto - precharge ON mode).

■ Structure of the Memory Setting Register (MCRB for FCRAM auto - precharge ON mode)

Settings for Memory configuration register (MCRB: Memory Configuration Register for extend type - B for FCRAM auto - precharge ON mode) is used to make various settings for FCRAM connected to the chip select area.

Figure 31.2-5 shows the bit configuration of the memory setting register (MCRB for FCRAM auto - precharge ON mode).

Figure 31.2-5 Structure of the Memory Setting Register (MCRB for FCRAM auto - precharge ON mode)



The register serves as the area for making various settings for FCRAM connected to the chip select area for which the access type (TYP3 to TYP0 bits) in the ACRn registers has been set as in Table 31.2-30 .

MB91460 Series

Table 31.2-30 lists the access type settings (TYP3 to TYP0 bits).

Table 31.2-21 Access type settings (TYP3 to TYP0 bits)

TYP3	TYP2	TYP1	TYP0	Access type
1	0	0	1	Memory type B: FCRAM (used auto precharge)

MCRB shares register hardware with MCRA. Updating the MCRB therefore updates the MCRA accordingly.

The functions are the same as MCRA. Note, however, that the function of the WBST bit is not available to this TYPE setting.

(FCRAM supports neither burst read nor single write mode.)

31.2.6 I/O Wait Registers for DMAC (IOWR0-3)

This section explains the configuration and functions of the I/O wait registers for DMAC (IOWR0-3).

■ Configuration of the I/O Wait Registers for DMAC (IOWR0-3)

The I/O wait registers for DMAC (IOWR0-3: I/O Wait Register for DMAC 0-3) set various kinds of waits during DMA fly-by access.

Figure 31.2-6 "Configuration of the I/O wait registers for DMAC (IOWR0, 1, 2, 3)" shows the configuration of the I/O wait registers for DMAC (IOWR0-3).

Figure 31.2-6 Configuration of the I/O Wait Registers for DMAC (IOWR0-3)

								Initial value		Access	
								INIT	RST		
IOWR0	31	30	29	28	27	26	25	24			
0000 0678 _H	RYE0	HLD0	WR01	WR00	IW03	IW02	IW01	IW00	xxxxxxxx _b	xxxxxxxx _b	W/R
IOWR1	23	22	21	20	19	18	17	16			
0000 0679 _H	RYE1	HLD1	WR11	WR10	IW13	IW12	IW11	IW10	xxxxxxxx _b	xxxxxxxx _b	W/R
IOWR2	15	14	13	12	11	10	9	8			
0000 067a _H	RYE2	HLD2	WR21	WR20	IW23	IW22	IW21	IW20	xxxxxxxx _b	xxxxxxxx _b	W/R
IOWR3	7	6	5	4	3	2	1	0			
0000 067b _H	RYE3	HLD3	WR31	WR30	IW33	IW32	IW31	IW30	xxxxxxxx _b	xxxxxxxx _b	W/R

■ Functions of Bits in the I/O Wait Registers for DMAC (IOWR0-3)

The following explains the functions of the bits in the I/O wait registers for DMAC.

[Bits 31, 23] RYEn,1 (RDY enable 0,1)

These bits set the wait control, using RDY, of channels 0-3 during DMAC fly-by access.

RYEn	RDY function setting
0	Disable RDY input for I/O access.
1	Enable RDY input for I/O access.

When 1 is set, wait insertion by the RDY pin can be performed during fly-by transfer on the relevant channel. IOWRX and IORDX are extended until the RDY pin is enabled. Also, RDX/WRX0-WRX3/WEX on the memory side are extended synchronously. If the chip select area of the fly-by transfer destination is set to RDY-enabled in the ACR register, wait insertion by the RDY pin can be performed regardless of the RYEn bit of IOWR. When the chip select area of the fly-by transfer destination is set to RDY-disabled in the ACR register, wait insertion by the RDY pin can only be performed during fly-by access if the area is set to RDY-enabled by the RYEn bit on the IOWR side.

[Bits 30,22] HLDn,1 (Hold Wait Control)

These bits control the hold cycle of the read strobe signal on the transfer source access side during DMA fly-by access.

HLDn	Hold wait setting
0	Do not insert a hold extension cycle.
1	Insert a hold extension cycle to extend the read cycle by one cycle.

If 0 is set, the read strobe signal (RDX for memory -> I/O and IORDX for I/O -> memory) and the write strobe signal (IOWRX for memory -> I/O and WRX0-WRX3 and WEX for I/O -> memory) on the transfer source access side are output at the same timing.

If 1 is set, the read strobe signal is output one cycle longer than the write strobe signal to secure a hold time for data at the transfer source access side when sending it to the transfer destination.

[Bits 29, 28, 21, 20] WR01/00, WR11/00 (I/O Idle Wait)

These bits set the number of idle cycles for continuous access during DMA fly-by access. [Table 31.2-22](#) "Settings for the Number of I/O Idle Cycles" lists the settings for the number of I/O idle cycles.

Table 31.2-22 Settings for the Number of I/O Idle Cycles

WRn1	WRn0	Setting of the number of I/O idle cycles
0	0	0 cycle
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

If one or more cycles is set as the number of idle cycles, cycles equal to the number specified are inserted after I/O access during DMA fly-by access. During the idle cycles, all CSX and strobe output is negated and the data pin is set to the high impedance state.

[Bits 27-24, 19-16, 11-8] IW03-00,IW13-10 (I/O Access Wait)

These bits set the number of auto-wait cycles for I/O access during DMA fly-by access.

MB91460 Series

Table 31.2-23 "Settings for the Number of I/O Wait Cycles" lists the settings for the number of I/O wait cycles.

Table 31.2-23 Settings for the Number of I/O Wait Cycles

IWn3	IWn2	IWn1	IWn0	Number of I/O wait cycles
0	0	0	0	0 cycle
0	0	0	1	1 cycle
...				...
1	1	1	1	15 cycle

Because data is synchronized between the transfer source and transfer destination, the I/O side setting of the IWnn bits and the wait setting for the fly-by transfer destination (such as memory), whichever is larger, is used as the number of wait cycles to be inserted. Consequently, more wait cycles than specified by the IWnn bits may be inserted.

31.2.7 Chip Select Enable Register (CSER)

Because data is synchronized between the transfer source and transfer destination, the I/O side setting of the IWnn bits and the wait setting for the fly-by transfer destination (such as memory), whichever is larger, is used as the number of wait cycles to be inserted. Consequently, more wait cycles than specified by the IWnn bits may be inserted.

■ Configuration of the Chip Select Enable Register (CSER)

The chip select enable register (CSER: Chip Select Enable register) enables and disables each chip select area.

Figure 31.2-7 "Configuration of the Chip Select Enable Register (CSER)" shows the configuration of the chip select enable register (CSER).

Figure 31.2-7 Configuration of the Chip Select Enable Register (CSER)

									Initial value		
	31	30	29	28	27	26	25	24	INIT	RST	Access
00000680 _H	CSE7	CSE6	CSE5	CSE4	CSE3	CSE2	CSE1	CSE0	00000001 _B	00000001 _B	R/W

■ Functions of Bits in the Chip Select Enable Register (CSER)

The following explains the functions of the bits in the chip select enable register (CSER).

[Bits 31-24] CSE7-0 (Chip Select Enable 0-7)

These bits are the chip select enable bits for CSX0-CSX7.

The initial value is 00000001_B, which enables only the CS0 area.

When 1 is written, a chip select area operates according to the settings of ASR0-7, ACR0-7, and AWR0-7.

Before setting this register, be sure to make all settings required for the corresponding chip select areas.

CSE7-0	Area control
0	Disable
1	Enable

Table 31.2-24 "CSXn Corresponding to the Chip Select Enable Bits" lists the corresponding CSXn for the chip select enable bits.

Table 31.2-24 CSXn Corresponding to the Chip Select Enable Bits

CSE bit	Corresponding CSXn
Bit 24: CSE0	CSX0
Bit 25: CSE1	CSX1
Bit 26: CSE2	CSX2
Bit 27: CSE3	CSX3
Bit 28: CSE4	CSX4
Bit 29: CSE5	CSX5
Bit 30: CSE6	CSX6
Bit 31: CSE7	CSX7

31.2.8 Cache Enable Register (CHER)

This section explains the configuration and functions of the cache enable register (CHER).

■ Configuration of the Cache Enable Register (CHER)

The cache enable register (CHER: CacHe Enable Register) controls the transfer of data read from each chip select area.

Figure 31.2-8 "Configuration of the Cache Enable Register (CHER)" shows the configuration of the cache enable register (CHER).

Figure 31.2-8 Configuration of the Cache Enable Register (CHER)

								Initial value			
								INIT	RST	Access	
	23	22	21	20	19	18	17	16			
00000681 _H	CHE7	CHE6	CHE5	CHE4	CHE3	CHE2	CHE1	CHE0	11111111 _B	11111111 _B	R/W

■ Functions of Bits in the Cache Enable Register (CHER)

The following explains the functions of the bits in the cache enable register (CHER).

[Bits 23-16] CHE7-0 (Cache Enable 7-0)

These bits enable and disable each chip select area for transfers to the built-in cache.

CHEn	Cache area setting
0	Not a cache area (data read from the applicable area is not saved in the cache)
1	Cache area (data read from the applicable area is saved in the cache)

MB91460 Series

31.2.9 Pin/Timing Control Register (TCR)

This section explains the configuration and functions of the pin/timing control register and its function.

■ Configuration of the Pin/Timing Control Register (TCR)

The pin/timing control register (TCR: Terminal and Limiting Control Register) controls the functions related to the general external bus interface controller, such as the setting of common pin functions and timing control.

Figure 31.2-9 "Configuration of the Pin/Timing Control Register (TCR)" shows the configuration of the pin/timing control register (TCR).

Figure 31.2-9 Configuration of the Pin/Timing Control Register (TCR)

									Initial value		Access
									INIT	RST	
	7	6	5	4	3	2	1	0			
00000683 _H	BREN	PSUS	PCLR	Reserved	Reserved	Reserved	RDW1	RDW0	00000000 _B	0000xxxx _B	R/W

■ Functions of Bits in the Pin/Timing Control Register (TCR)

The following explains the functions of the bits in the pin/timing control register (TCR).

[Bit 7] BREN (BRQ Enable)

This bit enables BRQ pin input and external bus sharing.

BREN	BRQ input enable setting
0	No bus sharing by BRQ/BGRNTX. BRQ input is disabled.
1	Bus sharing by BRQ/BGRNTX. BRQ input is enabled.

In the initial state (0), BRQ input is ignored. When 1 is set, the bus is made open (control with high impedance) and BGRNTX is activated (L level is output) when the bus is ready to be made open after the BRQ input becomes H level.

[Bit 6] PSUS (Prefetch suspend)

This bit controls temporary stopping of prefetch to all areas.

PSUS	Prefetch control
0	Enable prefetch
1	Suspend prefetch

If 1 is set, no new prefetch operation is performed before 0 is written. Since during this time the contents of the prefetch buffer are not deleted unless a prefetch buffer occurs, clear the prefetch buffer using the PCLR bit function (bit 5) before restarting prefetch.

[Bit 5] PCLR (Prefetch buffer clear)

This bit completely clears the prefetch buffer.

PCLR	Prefetch buffer control
0	Normal state
1	Clear the prefetch buffer.

If 1 is written, the prefetch buffer is cleared completely. When clearing is completed, the bit value automatically returns to 0. Interrupt (set to 1) the prefetch by the PSUS bit (bit 6) and then clear the buffer (It is also possible to write 11_B to both the PSUS and PCLR bits).

[Bit 4-2] Reserved

This bit is reserved. Be sure to set it to 0.

[Bits 1,0] RDW1,0 (Reduce Wait cycle)

These bits instruct all chip select areas and fly-by I/O channels to reduce only the number of auto-wait cycles in the auto-access cycle wait settings uniformly while the AWR register settings are retained unchanged. The settings for idle cycles, recovery cycles, setup, and hold cycles are not affected. [Table 31.2-25](#) "Settings for Wait Cycle Reduction" lists the settings for the wait cycle reduction for combinations of these bits.

Table 31.2-25 Settings for Wait Cycle Reduction

RDW1	RDW0	Wait cycle reduction
0	0	Normal wait (AWR0-7 settings)
0	1	1/2 (1-bit shift to the right) of the AWR0-7 settings
1	0	1/4 (2-bit shift to the right) of the AWR0-7 settings
1	1	1/8 (3-bit shift to the right) of the AWR0-7 settings

The purpose of this function is to prevent an excessive access cycle wait during operation on a low-speed clock (for example, when the base clock is switched to low speed or the frequency division ratio setting of the external bus clock is large).

To reset the wait cycle in these cases, each of the AWRs must usually be rewritten one at a time. However, when the RDW1/0 bit function is used, the access cycle wait is reduced for all of the AWRs in a single operation while all of the other high-speed clock settings in each register are retained.

Before returning the clock to high speed, be sure to reset the RDW1/0 bits to 00_B.

31.2.10 Refresh Control Register (RCR)

This section describes the bit configuration and functions of the refresh control register (RCR).

■ Structure of the Refresh Control Register (RCR)

The refresh control register (RCR) is used to make various refresh control settings for SDRAM.

The setting of this register is meaningless as long as SDRAM control is not set for any area, in that case the register value must not be updated from the initial state.

When read by a Read - modify - Write instruction, the SELF, RRLD, and PON bits always return to 0.

[Figure 31.2-10](#) shows the bit configuration of the refresh control register (RCR).

Figure 31.2-10 Structure of the Refresh Control Register (RCR)

RCRH bit		31	30	29	28	27	26	25	24	Initial value
Address	0000 0684 _H	SELF	RRLD	RFINT5	RFINT4	RFINT3	RFINT2	RFINT1	RFINT0	
		W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	00XXXXXX _B (INIT) 00XXXXXX _B (RST)
RCRL bit		23	22	21	20	19	18	17	16	Initial value
Address	0000 0685 _H	BRST	RFC2	RFC1	RFC0	PON	TRC2	TRC1	TRC0	
		W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	XXXX0XXX _B (INIT) XXXX0XXX _B (RST)

■ Bit Functions of the Refresh Control Register (RCR)

The following summarizes the functions of individual bits in the refresh control register (RCR).

[Bit 31] SELF (SELF refresh assert): Self - refresh control

This bit is used to control the self - refresh mode for memory that supports the self - refresh mode.

Table 31.2-26 lists the settings for self - refresh control.

Table 31.2-26 Settings for self - refresh control

SELF	Self - refresh control
0	Auto - refresh or power - down
1	Transition to self-refresh mode

Setting the bit to 1 performs a self - refresh after issuing the SELF command. Writing 0 terminates the self - refresh mode.

To hold the contents of SDRAM when putting the LSI into STOP state, use this bit to enter the self - refresh mode before entering the STOP state. At this time, centralized refreshing is performed before transition to the self - refresh mode. External access requests generated before it is completed are put on hold. The mode transits to the STOP state.

The device is released from the self - refresh mode either when 0 is written to this bit or access to SDRAM occurs. At this time, centralized refreshing is performed immediately after the release. If external access such as SDRAM access is attempted, therefore, the external access request is kept on hold and the CPU stops operation for a while. An attempt to put the LSI into the STOP state when it cannot enter the self - refresh mode causes it to directly enter the power save mode, resulting in corruption of data in SDRAM.

When read by a Read - modify - Write instruction, the SELF, RRLD, and PON bits always return to 0.

[Bit 30] RRLD (Refresh counter ReLoad): Refresh counter start control

This bit is used to start and reload the fresh counter.

Table 31.2-27 shows the function of refresh counter startup control.

Table 31.2-27 Function of refresh counter startup control

RRLD	Refresh counter startup control
0	Disable (no operation)
1	Execute auto - refreshing once and reload the RFINT value.

The refresh counter is inactive in the initial state.

If this bit is set to 1 in this state, all the SDRAM areas currently enabled in the CSER are auto - refreshed either once in distributed refresh mode or the RFC - specified number of times in centralized refresh mode. After that, the values in the RFINT5 to RFINT0 bits are reloaded.

From then on, the refresh counter starts being decremented. Whenever the counter causes an underflow from 000000_B, repeatedly, the values in the RFINT5 to RFINT0 bits are reloaded while at the same time auto - refreshing is performed once.

The bit returns to 0 upon completion of reloading.

To stop auto - refreshing, write 000000_B to the RFINT5 to RFINT0 bits.

When read by a Read - modify - Write instruction, the bit always returns a zero.

[Bits 29 - 24] RFINT5 to RFINT0 (ReFresh INTerval): Auto - refresh interval

Set these bits to the interval for automatic refreshing.

The auto - refresh interval can be obtained for distributed refresh mode $\{(RFINT5 - RFINT0 \text{ value}) \times 32 \times (\text{external bus clock cycle})\}$ or for centralized refresh mode $\{(RFINT5 - RFINT0 \text{ value}) \times 32 \times (\text{RFC specified number of times}) \times (\text{external bus clock cycle})\}$

Calculate the design value in consideration of the maximum RAS active time.

The refresh counter keeps on being decremented even while the auto - refresh command is being issued.

[Bit 23] BRST (BuRST refresh select): Burst refresh control

This bit is used to control the operation mode for auto - refreshing.

Table 31.2-28 shows the function of burst refresh control.

Table 31.2-28 Function of burst refresh control

BRST	Burst refresh control
0	Distributed refresh (Auto - refresh is activated at intervals.)
1	Burst refresh (Auto - refresh is activated repeatedly at one time.)

When distributed refreshing is set, the auto - refresh command is issued once at every refresh interval.

When burst refreshing is set, the auto - refresh command is issued continuously for the number of times set in the refresh counter at every refresh interval.

[Bits 22 - 20] RFC2, RFC1, RFC0 (ReFresh Count): Refresh count

Set these bits to the number of times a refresh must be performed to refresh all SDRAM.

Table 31.2-29 shows the number of times to refresh.

Table 31.2-29 Number of times to refresh

RFC2	RFC1	RFC0	Number of times to refresh
0	0	0	256
0	0	1	512
0	1	0	1024
0	1	1	2048
1	0	0	4096
1	0	1	8192
1	1	0	Setting prohibited
1	1	1	Refresh prohibited

The number of times to refresh specified here is the number of times centralized refreshing is performed before and after transition to the self - refresh mode. When burst refreshing has been selected with the BRST bit, the number of times to refresh is also the number of times the refresh command is issued at every refresh interval.

[Bit 19] PON (Power ON): Power - on control

This bit is used to control the SDRAM (FCRAM) power - on sequence.

Table 31.2-30 shows the function of power - on control.

Table 31.2-30 Function of power - on control

PON	Power-on control
0	Disabled (no-operation)
1	Start power-on sequence

Writing 1 to the PON bit starts the SDRAM power - on sequence.

Before starting the power - on sequence, be sure to set the relevant registers such as AWR, MCRA(B), and CSER.

This bit returns to 0 as soon as the power - on sequence is started.

When enabling the PON bit, set RFINT and enable RRLD to activate the refresh counter.

Refreshing is not performed only with the PON bit.

Do not enable this bit along with the SELF bit.

When read by a Read - modify - Write instruction, the bit always returns 0.

[Bits 18 - 16] TRC2, TRC1, TRC0 (Time of Refresh Cycle): Refresh cycle (tRC)

These bits set the refresh cycle (tRC).

Table 31.2-31 lists the settings for the refresh cycle (tRC).

Table 31.2-31 Settings for the refresh cycle (tRC)

TRC2	TRC1	TRC0	Refresh cycle (tRC)
0	0	0	4
0	0	1	5
0	1	0	6
0	1	1	7
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	11

31.3. Setting Example of the Chip Select Area

In the external bus interface, a total of eight chip select areas can be set.
This section presents an example of setting the chip select area.

■ Example of Setting the Chip Select Area

The address space of each area can be placed, in units of a minimum of 64 KB, anywhere in the 4 GB space using ASR0-7 (Area Select Registers) and ACR0-7 (Area Configuration Registers). When bus access is made to an area specified by these registers, the corresponding chip select signals (CSX0-CSX7) are activated (L output) during the access cycle.

● Example of setting ASRs and ASZ3-0

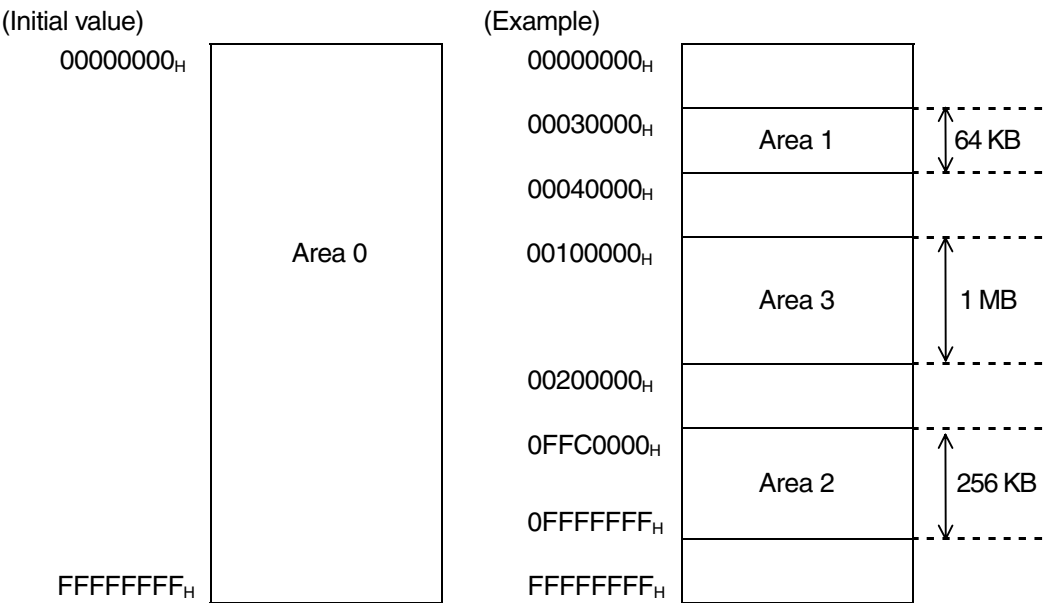
- ASR1=0003_H ACR1 ASZ3-0=0000_B: Chip select area 1 is assigned to 00030000_H to 0003FFFF_H.
- ASR2=0FFC_H ACR2 ASZ3-0=0010_B: Chip select area 2 is assigned to 0FFC0000_H to 10000000_H.
- ASR3=0011_H ACR3 SZ3-0=0100_B: Chip select area 3 is assigned to 00100000_H to 00200000_H.

Since at this point 1 MB is set for bits ASZ3-0 of the ACR, the unit for boundaries 1 MB and bits 19-16 of ASR3 are ignored. Before there is any writing to ACR0 after a reset, 00000000_H-FFFFFFFF_H is assigned to chip select area 0.

Set the chip select areas so that there is no overlap.

Figure 31.3-1 "Example of Setting the Chip Select Area" shows an example of setting the chip select area.

Figure 31.3-1 Example of Setting the Chip Select Area



MB91460 Series

31.4. Endian and Bus Access

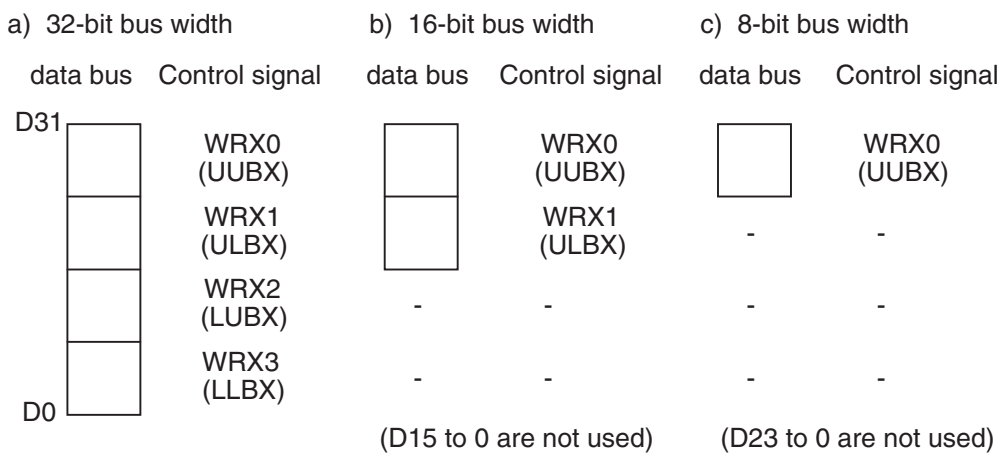
There is a one-to-one correspondence between the WRX0-WRX3 control signal and the byte location regardless of the endian method (big or little) and the data bus width. The following summarizes the location of bytes on the data bus of the MB91460 series used according to the specified data bus width and the corresponding control signal for each bus mode.

■ Relationship between Data Bus Width and Control Signal

This section summarizes the location of bytes on the data bus used according to the specified data bus width and the corresponding control signal for each bus mode.

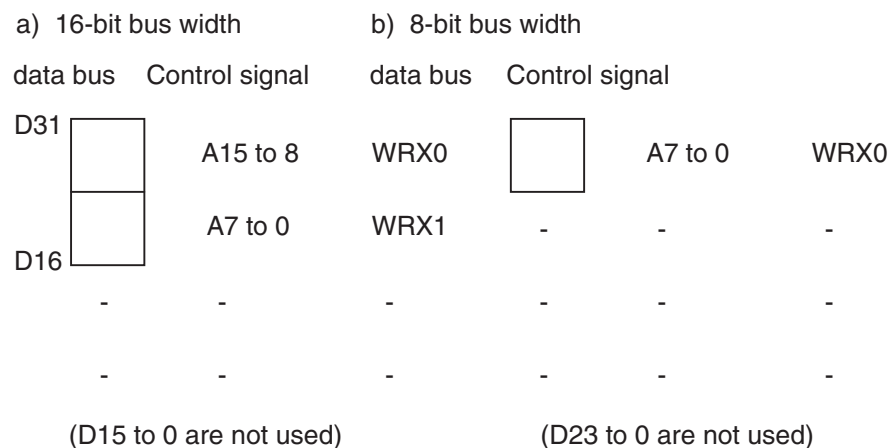
● Ordinary bus interface

Figure 31.4-1 Data Bus Width and Control Signal on the Ordinary Bus Interface



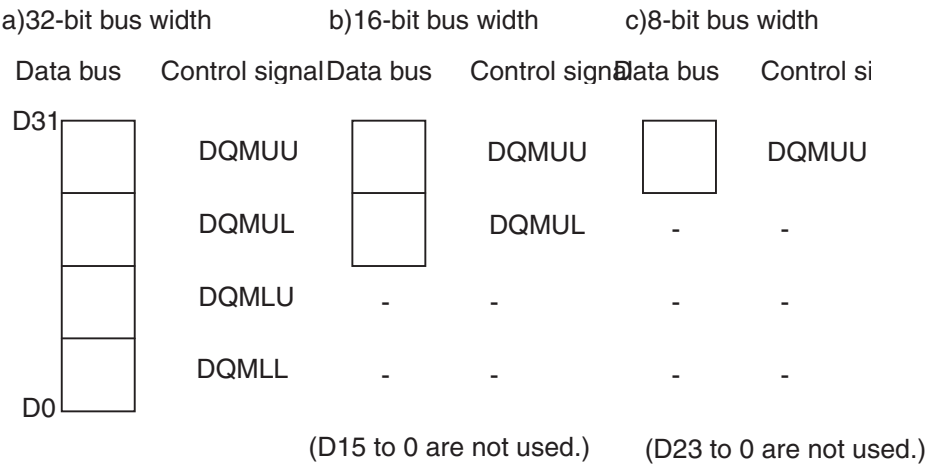
● Time division I/O interface

Figure 31.4-2 Data Bus Width and Control Signal in the Time Division I/O Interface



● SDRAM Interface

Figure 31.4-3 Data bus width of the SDRAM (FCRAM) interface and its control signals



31.4.1 Big Endian Bus Access

With the exception of the CS0 area of the MB91460 series, either the big endian method or the little endian method can be selected for each chip select area. If 0 is set for the LEND bit of the ACR register, the area is treated as big endian. The MB91460 series is normally big endian and performs external bus access.

■ Data Format

The relationship between the internal register and the external data bus is as follows:

- Word access (when LD/ST instruction executed)

Figure 31.4-4 Relationship between Internal Register and External Data Bus for Word Access

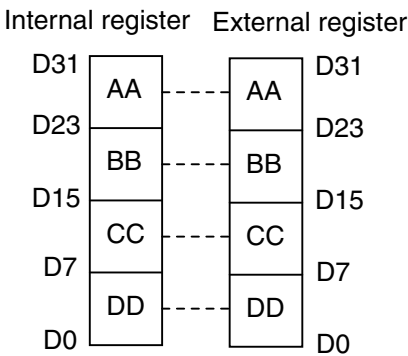


Figure 31.4-5 Relationship between the Internal Register and External Data Bus for Halfword Access

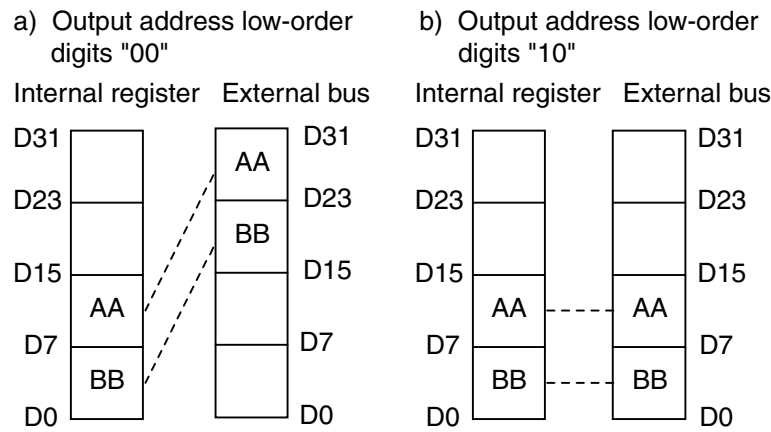
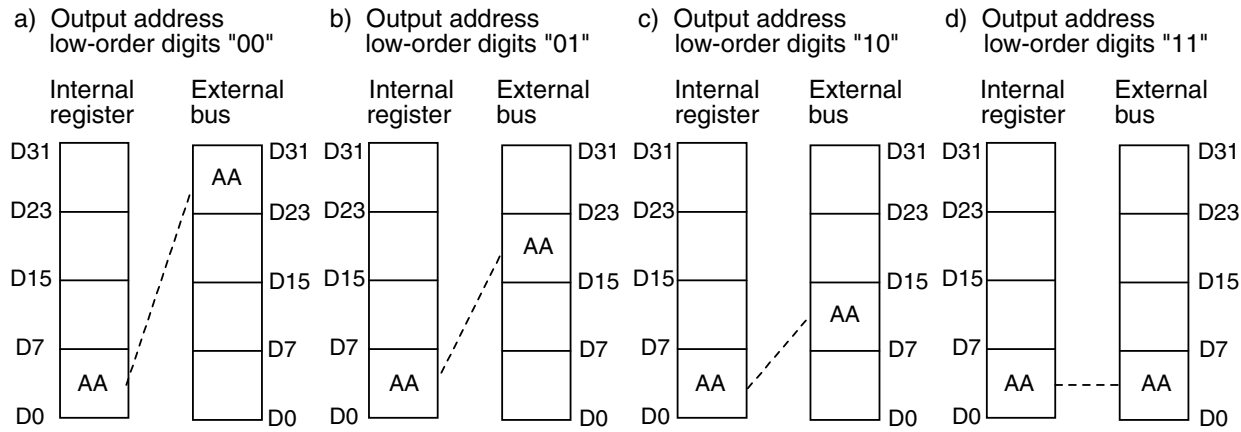


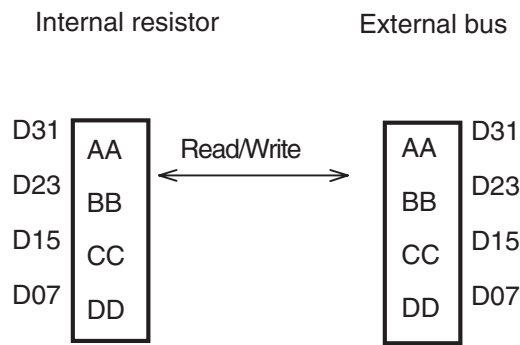
Figure 31.4-6 Relationship between Internal Register and External Data Bus for Byte Access



■ Data Bus Width

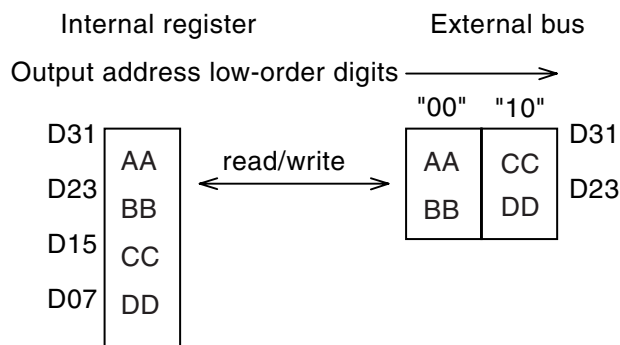
- 32-bit bus width

Figure 31.4-7 Relationship between Internal Register and External Bus Having 32-Bit Bus Width



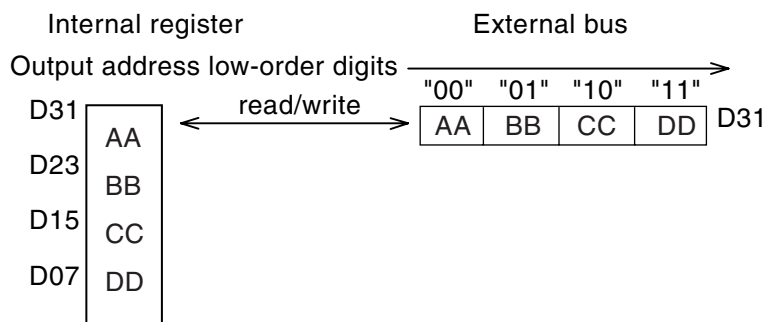
● 16-bit bus width

Figure 31.4-8 Relationship between Internal Register and External Bus Having 16-Bit Bus Width



● 8-bit bus width

Figure 31.4-9 Relationship between Internal Register and External Bus having 8-Bit bus Width



■ External Bus Access

Figure 31.4-11 "External bus Access for 16-Bit Bus Width" and Figure 31.4-12 "External Bus Access for 8-Bit Bus Width" show external bus access (16-bit/8-bit bus width) separately for word, halfword, and byte access. The following items are included in Figure 31.4-11 "External bus Access for 16-Bit Bus Width" and Figure 31.4-12 "External Bus Access for 8-Bit Bus Width":

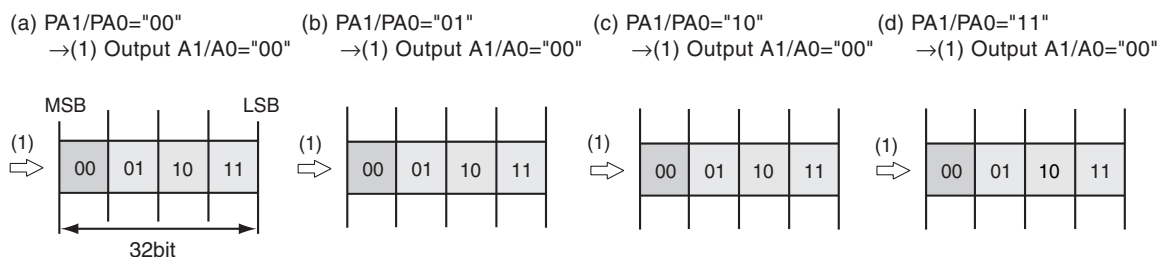
- Access byte location
- Program address and output address
- Bus access count

The MB91460 series does not detect misalignment errors.

Therefore, for word access, the lower two bits of the output address are always 00 regardless of whether 00, 01, 10, or 11 is specified as the lower two bits by the program. For halfword access, the lower two bits of the output address are 00 if the lower two bits specified by the program are 00 or 01, and are 10 if 10 or 11.

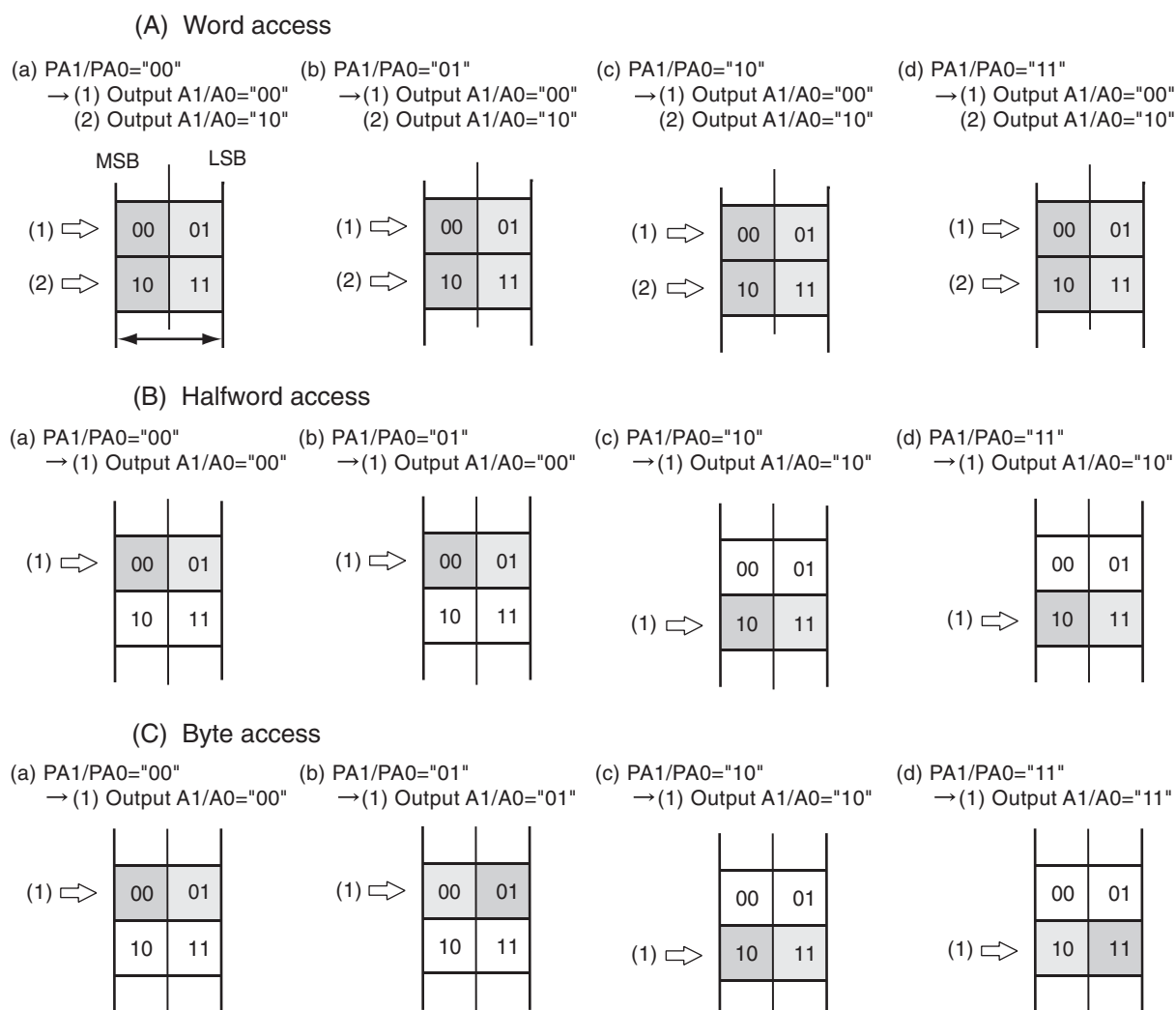
● 32-bit bus width

Figure 31.4-10 External bus Access for 32-Bit Bus Width



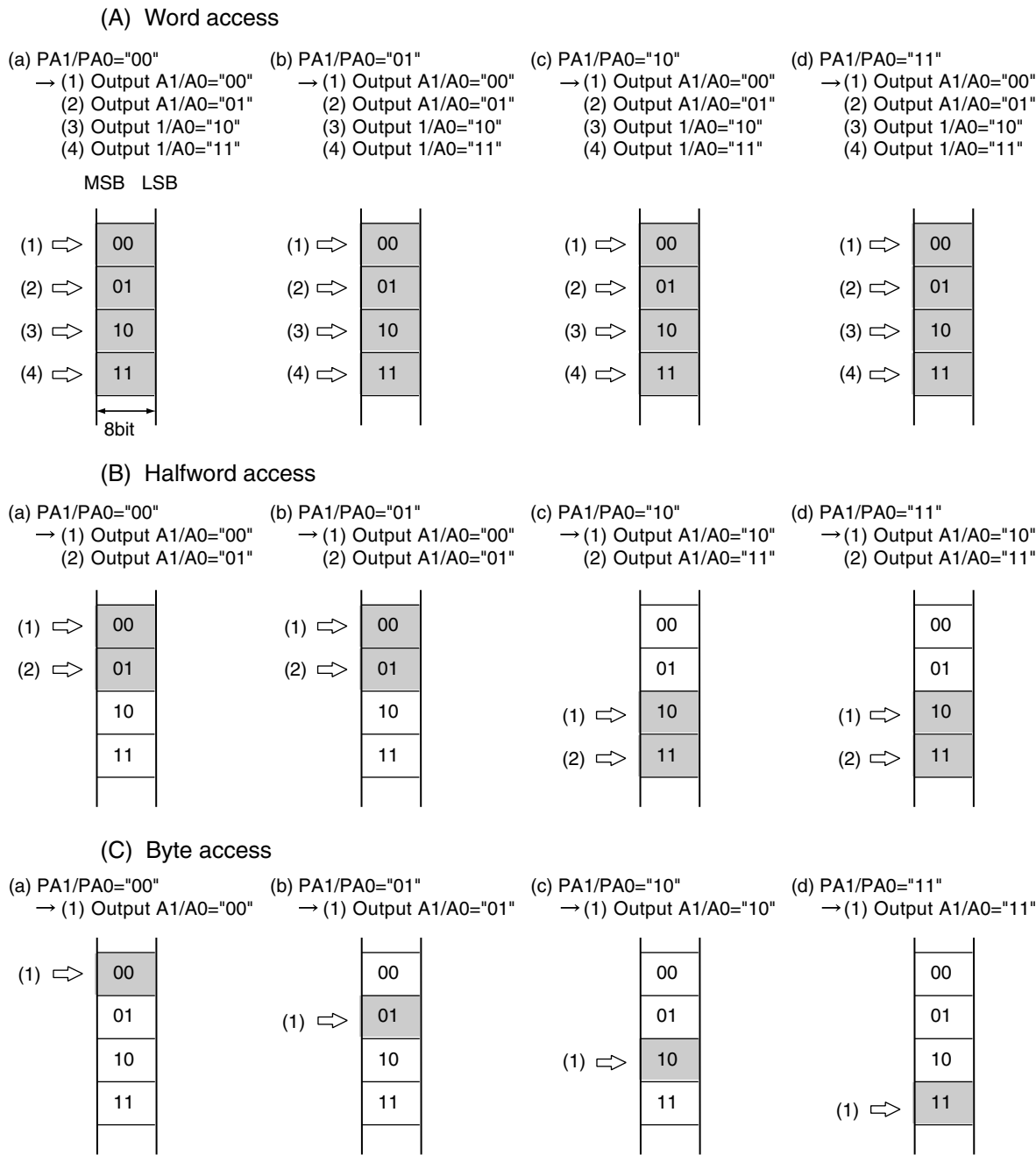
● 16-bit bus width

Figure 31.4-11 External bus Access for 16-Bit Bus Width



● 8-bit bus width

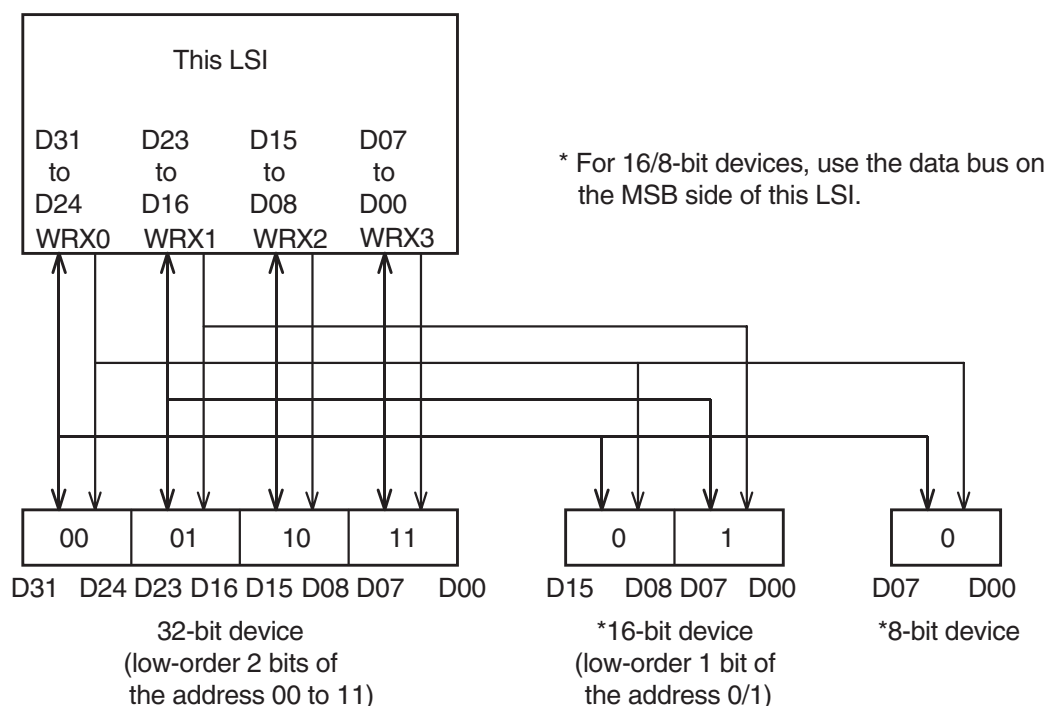
Figure 31.4-12 External Bus Access for 8-Bit Bus Width



■ Example of Connection with External Devices

Figure 31.4-13 "Example of Connecting the MB91460 Series to External Devices" shows an example of connection the MB91460 series to external devices.

Figure 31.4-13 Example of Connecting the MB91460 Series to External Devices



31.4.2 Little Endian Bus Access

Little endian (LER) external bus access is performed for an area for which the little endian method is set.

Little endian bus access on the MB91460 series is implemented by using the bus access operation used for the big endian method. Basically, the order of output addresses and control signal output are the same as for the big endian method and the byte locations on the data bus are swapped in accordance with the bus width.

Note that, when a connection is made, the big endian area and the little endian area must be kept physically separate.

■ Differences between Little Endian and Big Endian

The following explains the differences between little endian and big endian.

The order of addresses that are output is the same for little endian and big endian.

The data bus control signal used for 32/16/8-bit bus width is the same for little endian and big endian.

● Word access

The byte data on the MSB side for big endian address 00 becomes byte data on the LSB side when the little endian method is used.

For a word address, the locations of all four bytes in the word are reversed:

00 -> 11, 01 -> 10, 10 -> 01, 11 -> 00

● Halfword access

The byte data on the MSB side for the big endian address 0 becomes byte data on the LSB side when the little endian method is used.

For halfword access, the byte locations of two bytes are reversed.

0 -> 1, 1 -> 0

● Byte access

There is no difference between little endian and big endian.

■ Restrictions on the Little Endian Area

- If prefetch is enabled for a little endian area, always use word access to access the area. If data written to the prefetch buffer is accessed with any length other than word length, the correct endian conversion is not performed and the wrong data will be read. The reason is hardware restrictions related to the endian conversion mechanism.
- Do not place any instruction code in a little endian area.

■ Data Format

The relationship between the internal register and external data bus is as follows:

Figure 31.4-14 Relationship between the Internal Register and External Data Bus for Word Access

(1) Word access (when executing the LD/ST instructions)

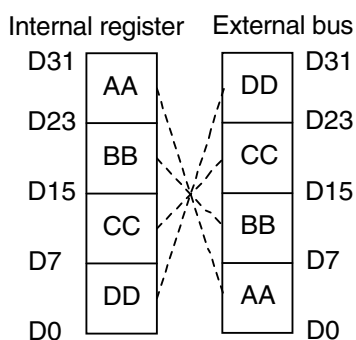


Figure 31.4-15 Relationship between Internal Register and External Data Bus for Halfword Access

(2) Halfword access (when executing the LDUH/STH instructions)

a) Output address low-order digits "00"

b) Output address low-order digits "10"

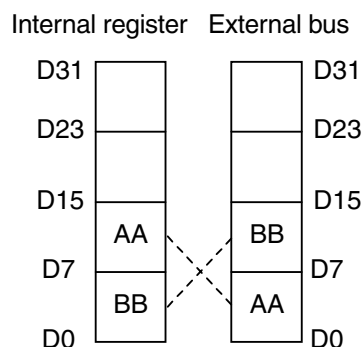
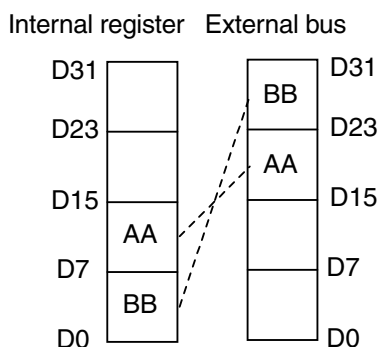
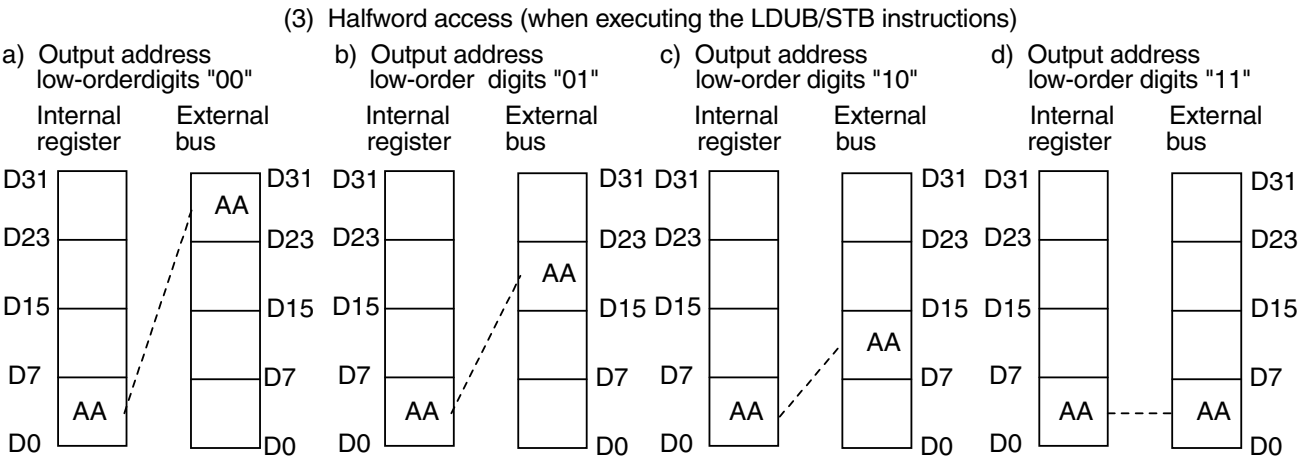


Figure 31.4-16 Relationship between Internal Register and External Data Bus for Byte Access

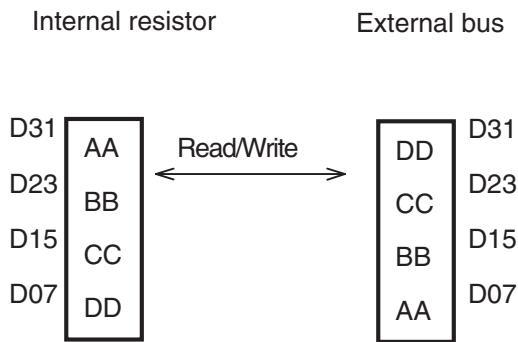


■ Data Bus Width

The following shows the relationships between the internal register and external data bus for each data bus width.

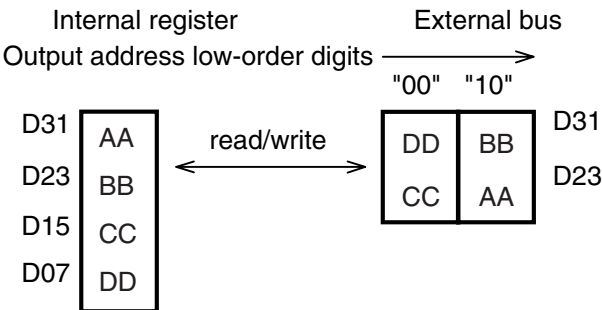
● 32-bit bus width

Figure 31.4-17 Relationship between Internal Register and External Bus Data for 32-bit Bus Width



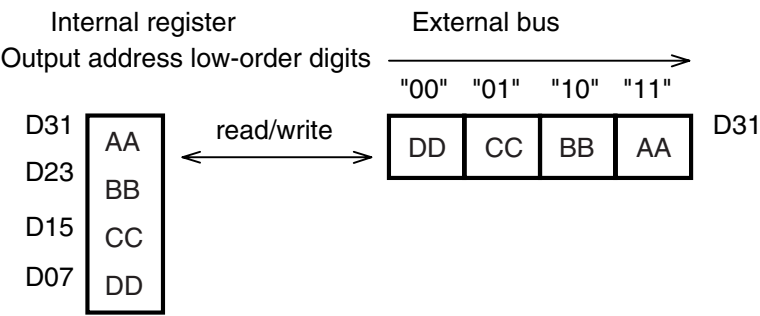
● 16-bit bus width

Figure 31.4-18 Relationship between Internal Register and External Bus Data for 16-bit Bus Width



● 8-bit bus width

Figure 31.4-19 Relationship between the Internal Register and External Data Bus in the 8-bit Bus Width

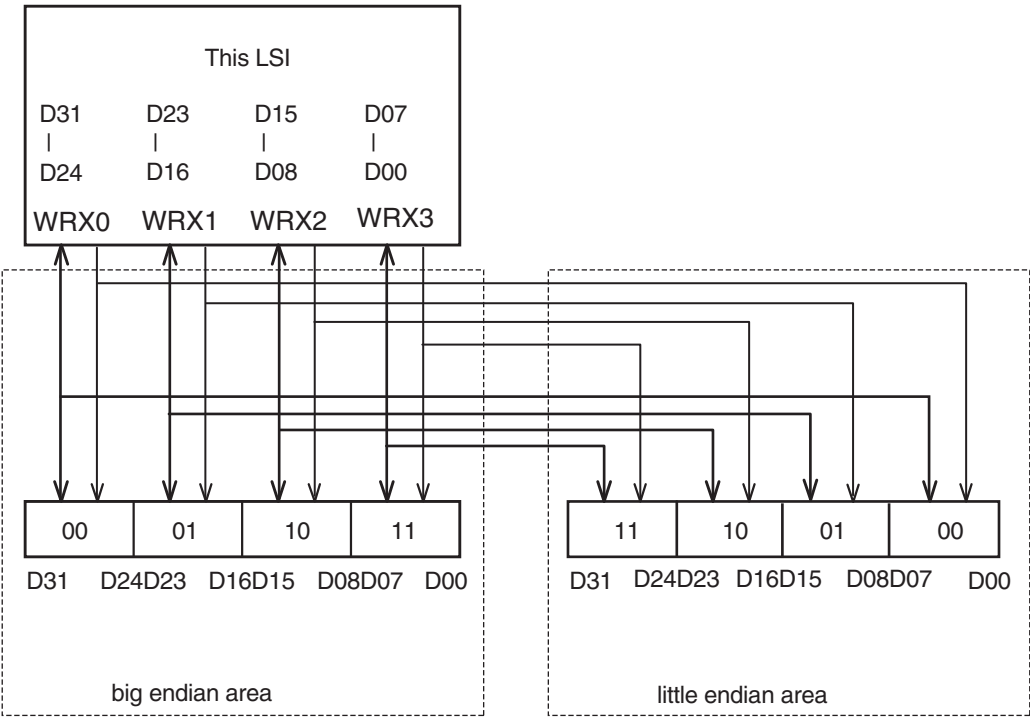


■ Examples of Connection with External Devices

The following shows examples of connecting the MB91460 series to external devices for each bus width.

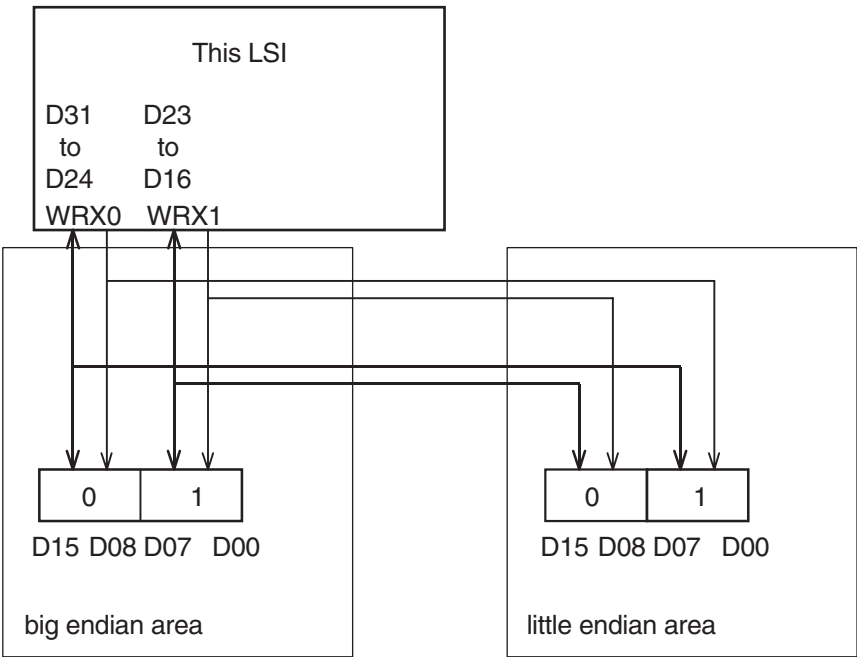
● 32-bit bus width

Figure 31.4-20 Example of Connecting the MB91460 Series to External Devices (32-Bit Bus Width)



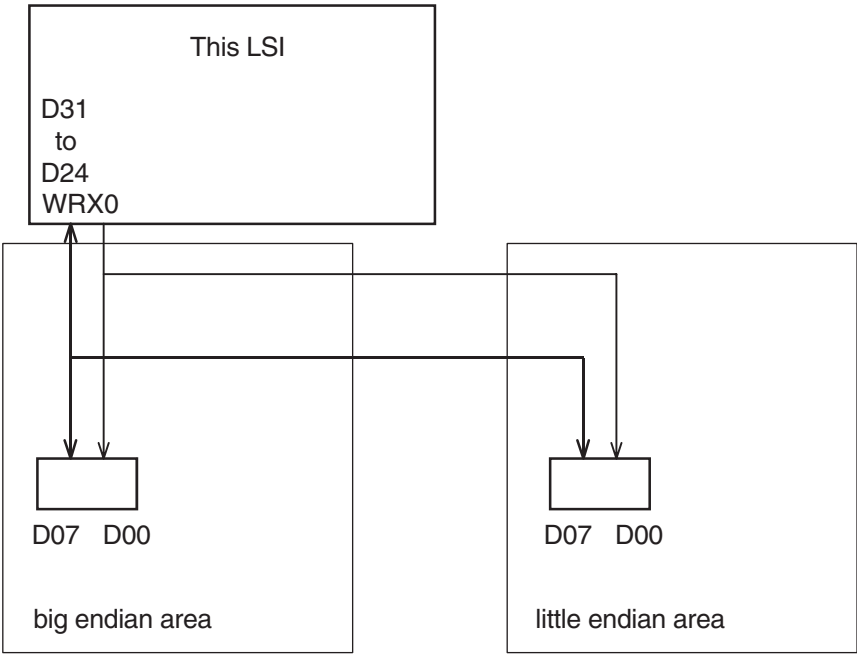
● 16-bit bus width

Figure 31.4-21 Example of Connecting the MB91460 Series to External Devices (16-Bit Bus Width)



● 8-bit bus width

Figure 31.4-22 Example of Connecting the MB91460 Series to External Devices (8-Bit Bus Width)



31.4.3 Comparison of Big Endian and Little Endian External Access

This section shows a comparison of big endian and little endian external access in word access, halfword access, and byte access for each bus width.

■ Word Access

Figure 31.4-23 Comparison of Big Endian and Little Endian External Access (Word access)

	Big endian mode	Little endian mode
32-bit bus width	<p>Internal Reg External terminal Control terminal</p> <p>address: Lower 2-bit: "0"</p> <p>(1)</p>	<p>Internal Reg External terminal Control terminal</p> <p>address : "0"</p> <p>(1)</p>
16-bit bus width	<p>Internal Reg External terminal Control terminal</p> <p>address: "0" "2"</p> <p>(1) (2)</p>	<p>Internal Reg External terminal Control terminal</p> <p>address: "0" "2"</p> <p>(1) (2)</p>
8-bit bus width	<p>Internal Reg External terminal Control terminal</p> <p>address: "0" "1" "2" "3"</p> <p>(1) (2) (3) (4)</p>	<p>Internal Reg External terminal Control terminal</p> <p>address: "0" "1" "2" "3"</p> <p>(1) (2) (3) (4)</p>

■ Halfword Access

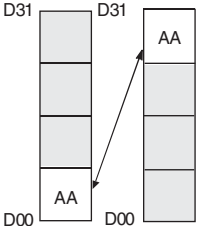
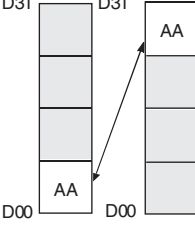
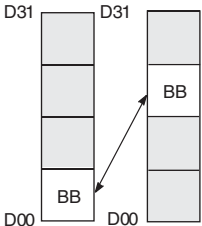
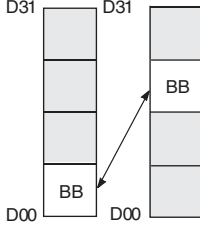
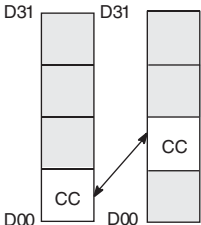
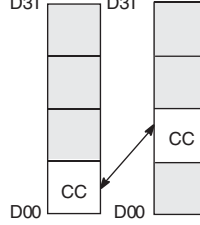
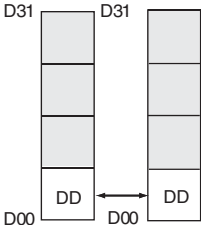
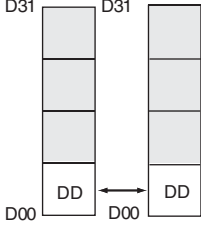
Figure 31.4-24 Comparison of Big Endian and Little Endian External Access (Halfword access)

	Big endian mode	Little endian mode
32-bit bus width	<div><div>Internal Reg address : "0"</div><div><div>D31</div><div>D00</div><div>AA</div><div>BB</div></div><div>External terminal address : "0"</div><div><div>D31</div><div>D00</div><div>AA</div><div>BB</div></div><div>Control terminal</div><div>WRX0</div><div>WRX1</div><div>-</div><div>-</div><div>(1)</div></div>	<div><div>Internal Reg address : "0"</div><div><div>D31</div><div>D00</div><div>AA</div><div>BB</div></div><div>External terminal address : "0"</div><div><div>D31</div><div>D00</div><div>BB</div><div>AA</div></div><div>Control terminal</div><div>WRX0</div><div>WRX1</div><div>-</div><div>-</div><div>(1)</div></div>
	<div><div>Internal Reg address : "2"</div><div><div>D31</div><div>D00</div><div>CC</div><div>DD</div></div><div>External terminal address : "2"</div><div><div>D31</div><div>D00</div><div>CC</div><div>DD</div></div><div>Control terminal</div><div>-</div><div>-</div><div>WRX2</div><div>WRX3</div><div>(1)</div></div>	<div><div>Internal Reg address : "2"</div><div><div>D31</div><div>D00</div><div>CC</div><div>DD</div></div><div>External terminal address : "2"</div><div><div>D31</div><div>D00</div><div>DD</div><div>CC</div></div><div>Control terminal</div><div>-</div><div>-</div><div>WRX2</div><div>WRX3</div><div>(1)</div></div>

	Big endian mode	Little endian mode
16-bit bus width	<p>Internal Reg address: "0"</p> <p>D31 D00</p> <p>External terminal address: "0"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>WRX1</p> <p>—</p> <p>—</p> <p>(1)</p>	<p>Internal Reg address: "0"</p> <p>D31 D00</p> <p>External terminal address: "0"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>WRX1</p> <p>—</p> <p>—</p> <p>(1)</p>
	<p>Internal Reg address: "2"</p> <p>D31 D00</p> <p>External terminal address: "2"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>WRX1</p> <p>—</p> <p>—</p> <p>(1)</p>	<p>Internal Reg address: "2"</p> <p>D31 D00</p> <p>External terminal address: "2"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>WRX1</p> <p>—</p> <p>—</p> <p>(1)</p>
8-bit bus width	<p>Internal Reg address: "0" "1"</p> <p>D31 D00</p> <p>External terminal address: "0" "1"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>—</p> <p>—</p> <p>—</p> <p>(1) (2)</p>	<p>Internal Reg address: "0" "1"</p> <p>D31 D00</p> <p>External terminal address: "0" "1"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>—</p> <p>—</p> <p>—</p> <p>(1) (2)</p>
	<p>Internal Reg address: "2" "3"</p> <p>D31 D00</p> <p>External terminal address: "2" "3"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>—</p> <p>—</p> <p>—</p> <p>(1) (2)</p>	<p>Internal Reg address: "2" "3"</p> <p>D31 D00</p> <p>External terminal address: "2" "3"</p> <p>D31 D00</p> <p>Control terminal</p> <p>WRX0</p> <p>—</p> <p>—</p> <p>—</p> <p>(1) (2)</p>

■ Byte Access

Figure 31.4-25 Comparison of Big Endian and Little Endian External Access (Byte access)

	Big endian mode	Little endian mode
32-bit bus width	<p>Internal Reg address : "0" External terminal address : "0" Control terminal</p>  <p>(1)</p>	<p>Internal Reg address : "0" External terminal address : "0" Control terminal</p>  <p>(1)</p>
	<p>Internal Reg address : "1" External terminal address : "1" Control terminal</p>  <p>(1)</p>	<p>Internal Reg address : "1" External terminal address : "1" Control terminal</p>  <p>(1)</p>
	<p>Internal Reg address : "2" External terminal address : "2" Control terminal</p>  <p>(1)</p>	<p>Internal Reg address : "2" External terminal address : "2" Control terminal</p>  <p>(1)</p>
	<p>Internal Reg address : "3" External terminal address : "3" Control terminal</p>  <p>(1)</p>	<p>Internal Reg address : "3" External terminal address : "3" Control terminal</p>  <p>(1)</p>

	Big endian mode	Little endian mode
16-bit bus width	<div><div>Internal Reg</div><div>address: "0"</div><div><div>D31</div><div>D31</div><div>D16</div><div>AA</div><div>D00</div></div><div><div>External terminal</div><div>AA</div></div><div><div>Control terminal</div><div>WRX0</div><div>—</div><div>—</div><div>—</div></div><div>(1)</div></div>	<div><div>Internal Reg</div><div>address: "0"</div><div><div>D31</div><div>D31</div><div>D16</div><div>AA</div><div>D00</div></div><div><div>External terminal</div><div>AA</div></div><div><div>Control terminal</div><div>WRX0</div><div>—</div><div>—</div><div>—</div></div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>address: "1"</div><div><div>D31</div><div>D31</div><div>D16</div><div>BB</div><div>D00</div></div><div><div>External terminal</div><div>BB</div></div><div><div>Control terminal</div><div>—</div><div>WRX1</div><div>—</div><div>—</div></div><div>(1)</div></div>	<div><div>Internal Reg</div><div>address: "1"</div><div><div>D31</div><div>D31</div><div>D16</div><div>BB</div><div>D00</div></div><div><div>External terminal</div><div>BB</div></div><div><div>Control terminal</div><div>—</div><div>WRX1</div><div>—</div><div>—</div></div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>address: "2"</div><div><div>D31</div><div>D31</div><div>D16</div><div>CC</div><div>D00</div></div><div><div>External terminal</div><div>CC</div></div><div><div>Control terminal</div><div>WRX0</div><div>—</div><div>—</div><div>—</div></div><div>(1)</div></div>	<div><div>Internal Reg</div><div>address: "2"</div><div><div>D31</div><div>D31</div><div>D16</div><div>CC</div><div>D00</div></div><div><div>External terminal</div><div>CC</div></div><div><div>Control terminal</div><div>WRX0</div><div>—</div><div>—</div><div>—</div></div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>address: "3"</div><div><div>D31</div><div>D31</div><div>D16</div><div>DD</div><div>D00</div></div><div><div>External terminal</div><div>DD</div></div><div><div>Control terminal</div><div>—</div><div>WRX1</div><div>—</div><div>—</div></div><div>(1)</div></div>	<div><div>Internal Reg</div><div>address: "3"</div><div><div>D31</div><div>D31</div><div>D16</div><div>DD</div><div>D00</div></div><div><div>External terminal</div><div>DD</div></div><div><div>Control terminal</div><div>—</div><div>WRX1</div><div>—</div><div>—</div></div><div>(1)</div></div>

	Big endian mode	Little endian mode
8-bit bus width	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "0"</div><div><div>D31</div><div>D24</div><div>AA</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "0"</div><div><div>D31</div><div>D24</div><div>AA</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "1"</div><div><div>D31</div><div>D24</div><div>BB</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "1"</div><div><div>D31</div><div>D24</div><div>BB</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "2"</div><div><div>D31</div><div>D24</div><div>CC</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "2"</div><div><div>D31</div><div>D24</div><div>CC</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>
	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "3"</div><div><div>D31</div><div>D24</div><div>DD</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>	<div><div>Internal Reg</div><div>External terminal</div><div>Control terminal</div><div>address: "3"</div><div><div>D31</div><div>D24</div><div>DD</div><div>D00</div></div><div>WRX0</div><div>—</div><div>—</div><div>—</div><div>(1)</div></div>

31.5. Operation of the Ordinary bus interface

This section explains operation of the ordinary bus interface.

■ Ordinary Bus Interface

For the ordinary bus interface, two clock cycles are the basic bus cycles for both read access and write access.

The following operational phases of the ordinary bus interface are explained below with the use of a timing chart.

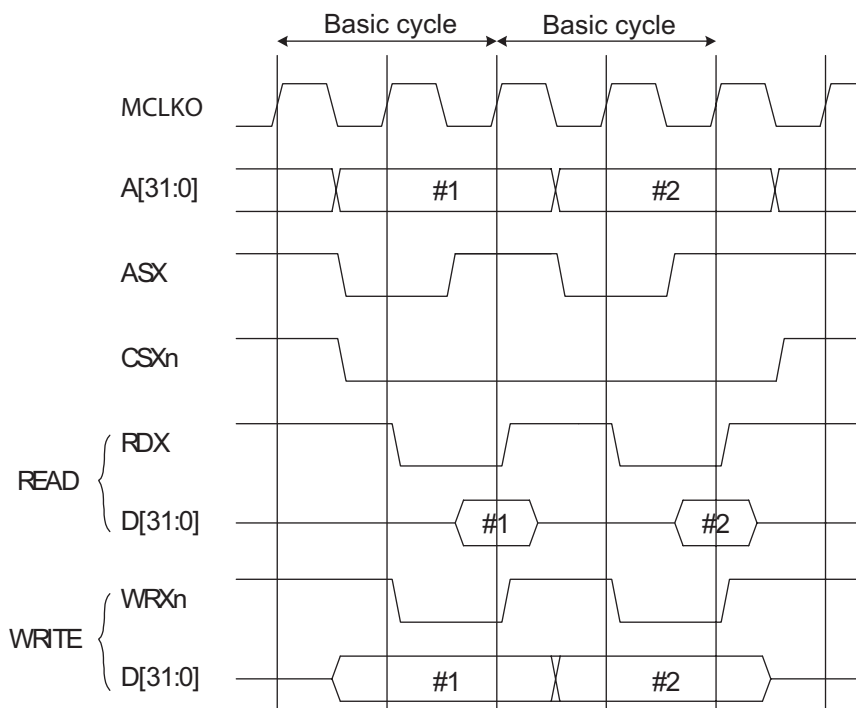
- Basic timing (for successive accesses)
- WRXn + byte control type
- Read -> write
- Write -> write
- Auto-wait cycle
- External wait cycle
- Synchronous write enable output
- CSXn delay setting
- CSXn -> RDX/WRXn/WEX setup, RDX/WRXn/WEX -> CSXn hold setting
- DMA fly-by transfer (I/O -> memory)
- DMA fly-by transfer (memory -> I/O)

31.5.1 Basic Timing

This section shows the basic timing for successive accesses.

■ Basic Timing (For Successive Accesses)

Figure 31.5-1 "Basic Timing (For Successive Accesses)" shows the operation timing for (TYP3-0 = 0000_B, AWR = 0008_H)

Figure 31.5-1 Basic Timing (For Successive Accesses)

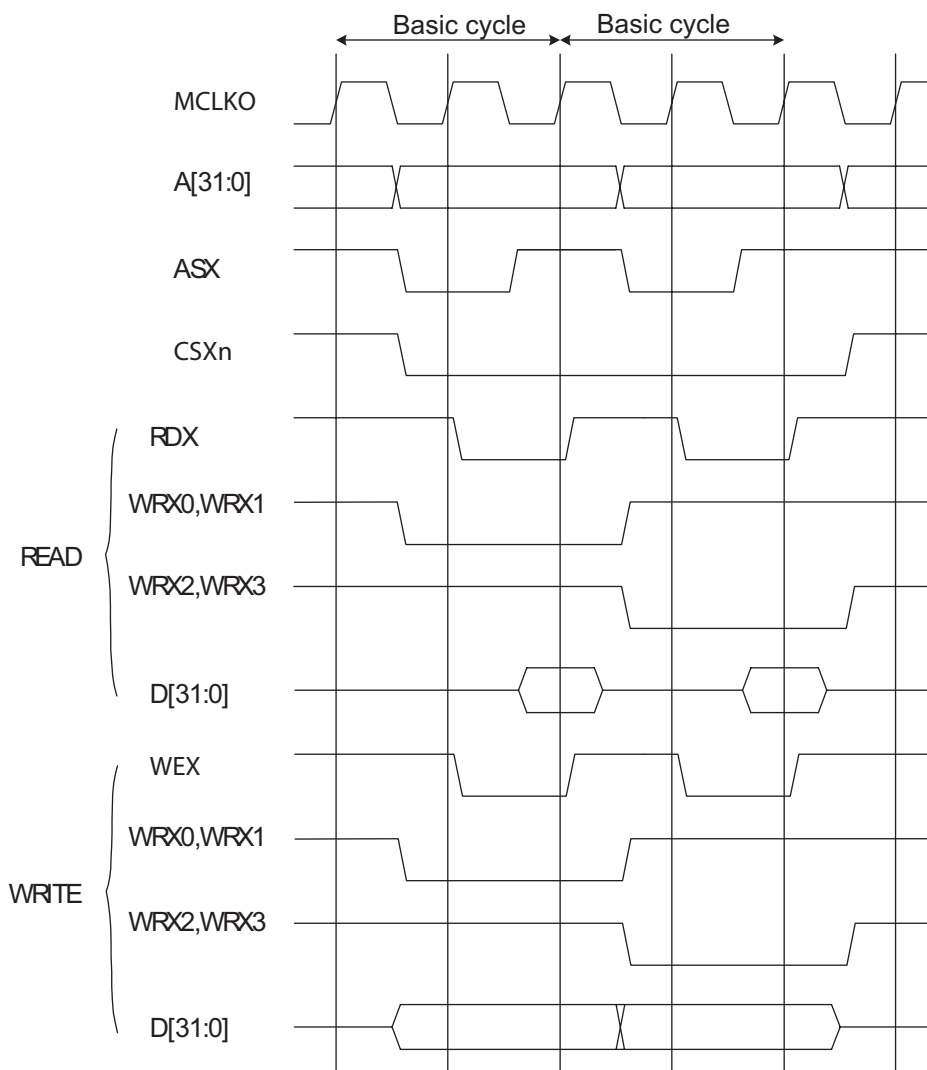
- ASX is asserted for one cycle in the bus access start cycle.
- A31-0 continues to output the address of the location of the start byte in word/halfword/byte access from the bus access start cycle to the bus access end cycle.
- If the W02 bit of the AWR0-7 registers is 0, CSX0-CSX7 are asserted at the same timing as ASX. For successive accesses, CSX0-CSX7 are not negated. If the W00 bit of the AWR register is 0, CSX0-CSX7 are negated after the bus cycle ends. If the W00 bit is 1, CSX0-CSX7 are negated one cycle after bus access ends.
- RDX and WRX0-WRX3 are asserted from the 2nd cycle of the bus access. Negation occurs after the wait cycle of bits W15-W12 of the AWR register is inserted. The timing of asserting RDX and WRX0-WRX3 can be delayed by one cycle by setting the W01 bit of the AWR register to 1. However, depending on the internal state, the assertion of WRX0-WRX3 may not start in the 2nd cycle and may even be delayed if the W01 bit is set to 0.
- If a setting is made so that WRX0-WRX3 is used like TYP3-0=0x0xB, WEX is always H.
- For read access, D31-0 is read when MCLKO rises in the cycle in which the wait cycle ended after RDX was asserted.
- For write access, data output to D31-0 starts at the timing at which WRX0-WRX3 are asserted.

31.5.2 Operation of WRXn + Byte Control Type

This section shows the operation timing for the WRXn + byte control type.

■ Operation Timing of the WRXn + Byte Control Type

Figure 31.5-2 "Timing Chart for the WRXn + Byte Control Type" shows the operation timing for (TYP3-0 = 0010_B, AWR = 0008_H).

Figure 31.5-2 Timing Chart for the \overline{WRn} + Byte Control Type

- Operation of ASX, CSXn, RDX, A31-0, and D31-16 is the same as that described in 31.5.1 "Basic Timing". WEX is asserted from the 2nd cycle of the bus access. Negation occurs after the wait cycle of bits W15-W12 of the AWR register is inserted. The timing of asserting RDX and WRX0-WRX3 can be delayed by one cycle by setting the W01 bit of the AWR register to 1. However, depending on the internal state, assertion of WRX0-WRX3 may not start in the 2nd cycle and may even be delayed if the W01 bit is set to 0. (Operation is the same as that for WRX0-WRX3 described in 31.5.1 "Basic Timing" .)
- WRX0-WRX3 indicate the byte location expressed with negative logic when they are used for access as the byte enable signal. Assertion continues from the bus access start cycle to the bus access end cycle and changes at the same timing as the address timing. The byte location for access is indicated for both read access and write access.
- For write access, data output to D31-16 starts at the timing at which WRXn is asserted. If the areas defined by TYP3-0=0x0xB (WRX0-WRX3 used) and TYP3-0=0x1xB (WRXn + byte control) are mixed, be sure to make the following setting for all areas that will be used. (For details, see the notes).
 - Set at least one read -> write idle cycle.
 - Set at least one write recovery cycle.

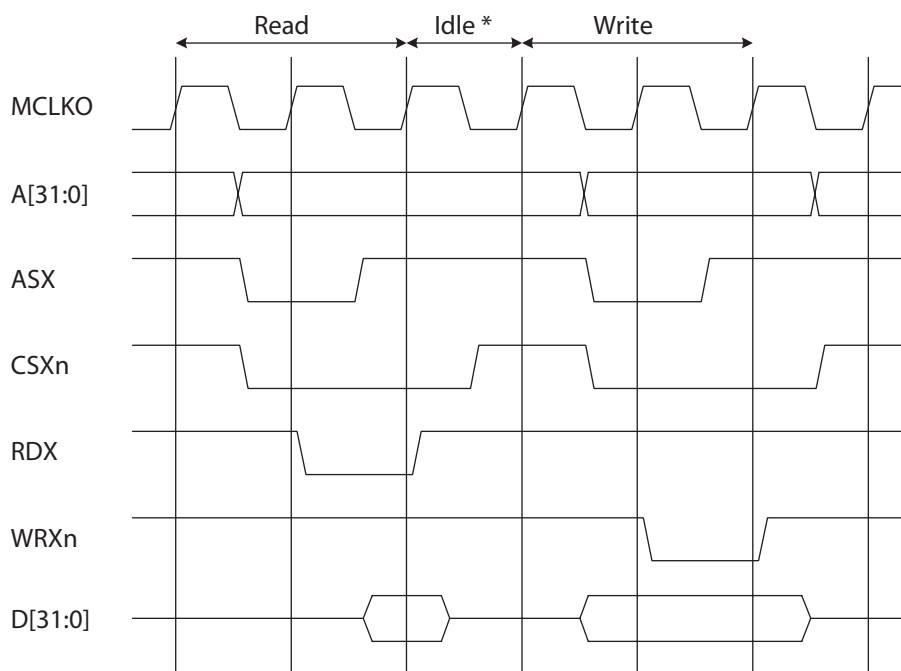
31.5.3 Read -> Write Operation

This section shows the operating timing for read -> write.

■ Operation Timing of Read -> Write

Figure 31.5-3 "Timing Chart for Read -> Write" shows the operation timing for (TYP3-0=0000_B, AWR=0048_H).

Figure 31.5-3 Timing Chart for Read -> Write



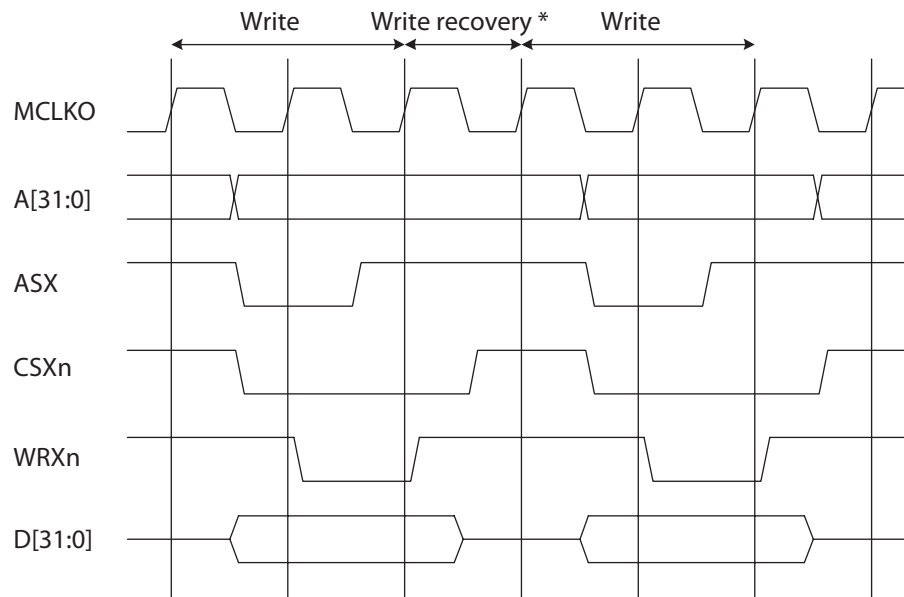
- *) Setting of the W07/W06 bits of the AWR register enables 0-3 idle cycles to be inserted.
- Settings in the CS area on the read side are enabled.
- This idle cycle is inserted if the next access after a read access is write access or access to another area.

31.5.4 Write -> Write Operation

This section shows the operation timing for write -> write.

■ Write -> Write Operation

Figure 31.5-4 "Timing Chart for the Write -> Write Operation" shows the operation timing for (TYP3-0=0000_B, WR=0018_H).

Figure 31.5-4 Timing Chart for the Write -> Write Operation

- *) Setting of the W05/W04 bits of the AWR register enables 0-3 write cycles to be inserted.
- After all of the write cycles, recovery cycles are generated.
- Write recovery cycles are also generated if write access is divided into phases for access with a bus width wider than that specified.

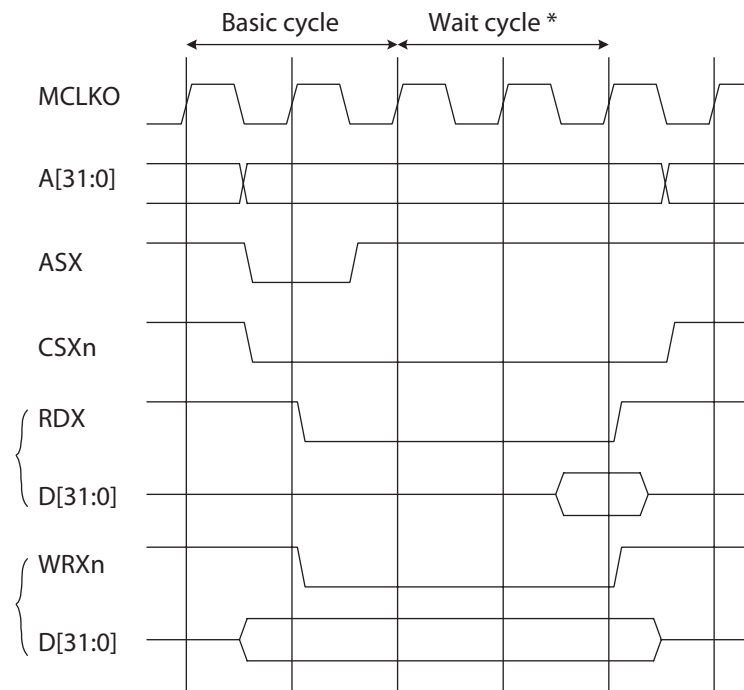
31.5.5 Auto-Wait Cycle

This section shows the operation timing for the auto-wait cycle.

■ Auto-Wait Cycle Timing

Figure 31.5-5 "Timing Chart for the Auto-Wait Cycle" shows the operation timing for (TYP3-0=0000_B, AWR=2008_H).

Figure 31.5-5 Timing Chart for the Auto-Wait Cycle



*) Setting of the W15-12 bits (first wait cycles) of the AWR register enables 0-15 auto-wait cycles to be set.

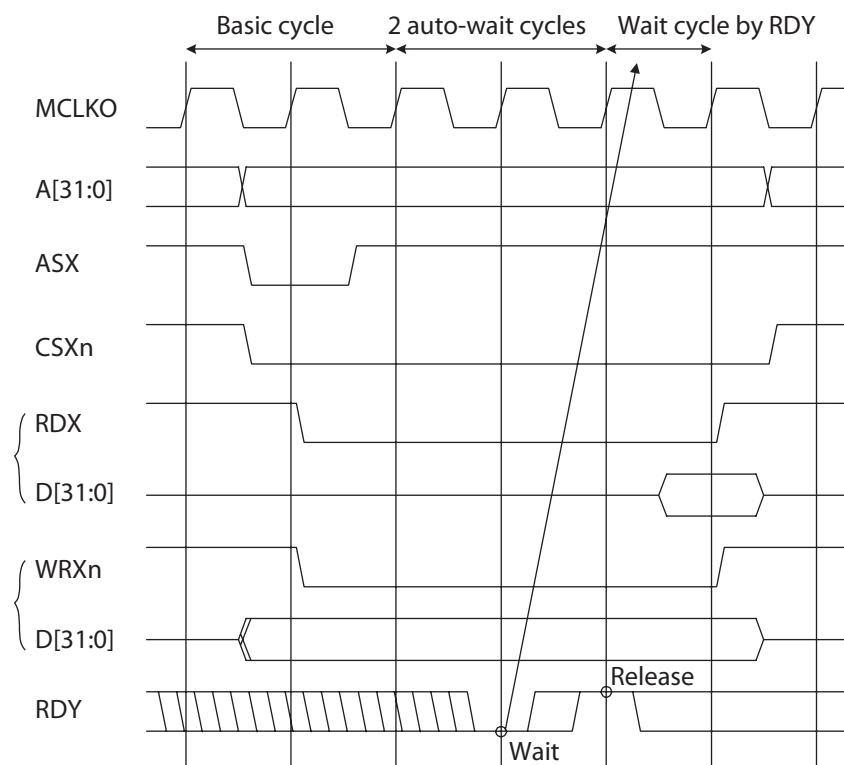
In Figure 31.5-5 "Timing Chart for the Auto-Wait Cycle", two auto-wait cycles are inserted, making a total of four cycles for access. If auto-wait is set, the minimum number of bus cycles is 2 cycles + (first wait cycles). For a write operation, the minimum number of bus cycles may be still longer depending on the internal state.

31.5.6 External Wait Cycle

This section shows the operation timing for the external wait cycle.

■ External Wait Cycle Timing

Figure 31.5-6 "Timing Chart for the External Wait Cycle" shows the operation timing for (TYP3-0=0001_B, AWR=2008_H).

Figure 31.5-6 Timing Chart for the External Wait Cycle

Setting 1 for the TYP0 bit of the ACR register and enabling the external RDY input pin enables external wait cycles to be inserted.

In [Figure 31.5-6](#), the oblique-lined portion of the RDY pin is invalid because the wait based on the automatic wait cycle remains in effect.

The value at the RDY input pin is evaluated from the last automatic wait cycle on.

Once a wait cycle is completed, the value at the RDY input pin remains invalid until the next access cycle is started.

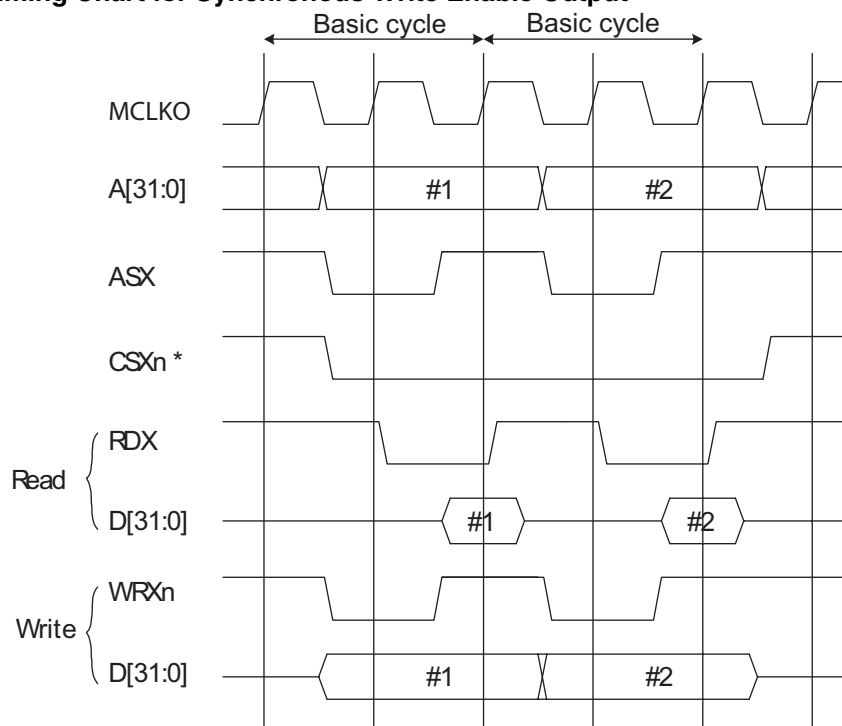
31.5.7 Synchronous Write Enable Output

This section shows the operation timing for synchronous write enable output.

■ Operation Timing for Synchronous Write Enable Output

[Figure 31.5-7](#) "Timing Chart for Synchronous Write Enable Output" shows the operation timing for (TYP3-0=0000_B, AWR=0000_H).

Figure 31.5-7 Timing Chart for Synchronous Write Enable Output



- If synchronous write enable output is enabled (If the W03 bit of the AWR is 1), operation is as follows.
- WRX0-WRX3 and WEX pin output asserts synchronous write enable output at the timing at which ASX pin output is asserted. For a write to an external bus, the synchronous write enable output is L. For a read from an external bus, the synchronous write enable output is H.
- Write data is output from the external data output pin in the clock cycle following the cycle in which synchronous write enable output is asserted. If write data cannot be output because the internal bus is temporarily unavailable, assertion of synchronous write enable output may be extended until write data can be output.
- Read strobe output (RDX) functions as an asynchronous read strobe regardless of the setting of WRX0-WRX3 and WEX output timing. Use it as is for controlling the data I/O.
- If synchronous write enable output is used, the following restrictions apply:

Do not set the following additional wait because the timing for synchronous write enable output becomes meaningless:

- CSX -> RDX/WRXn/WEX setup (Always write 0 to the W01 bit of AWR)
- First wait cycle setting (Always write 0000 to bits W15-W12 of AWR)

Do not set the following access types (TYPE3-0 bits (Bits 3-0) in the ACR register) because the timing for synchronous write enable output becomes meaningless:

- Multiplex bus setting (Always write 0 to the TYPE2 bit of ACR)
- RDY input enable setting (Always write 0 to the TYPE0 bit of ACR)

Always set the burst length to "1" (BST1 to 0 bit = 0) for the synchronous write enable output

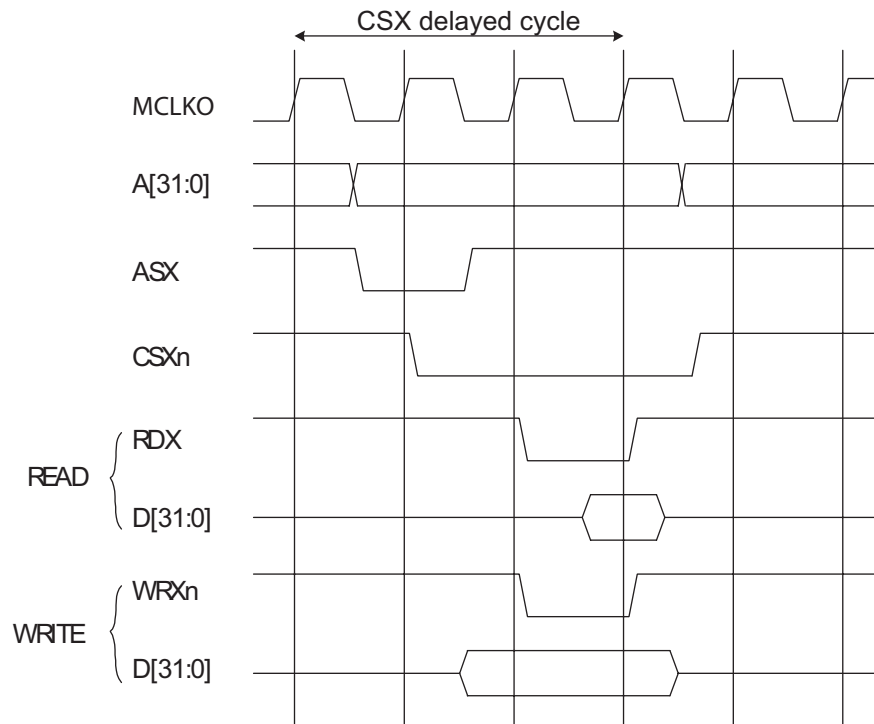
31.5.8 CSXn Delay Setting

This section shows the operation timing for the CSXn delay setting.

■ Operation Timing for the CSX Delay Setting

Figure 31.5-8 "Operation Timing Chart for the CSX Delay Setting" shows the operation timing for (TYP3-0=0000_B, AWR=000C_H).

Figure 31.5-8 Operation Timing Chart for the CSX Delay Setting



If the W02 bit is 1, assertion starts in the cycle following the cycle in which ASX is asserted. For successive accesses, a negation period is inserted.

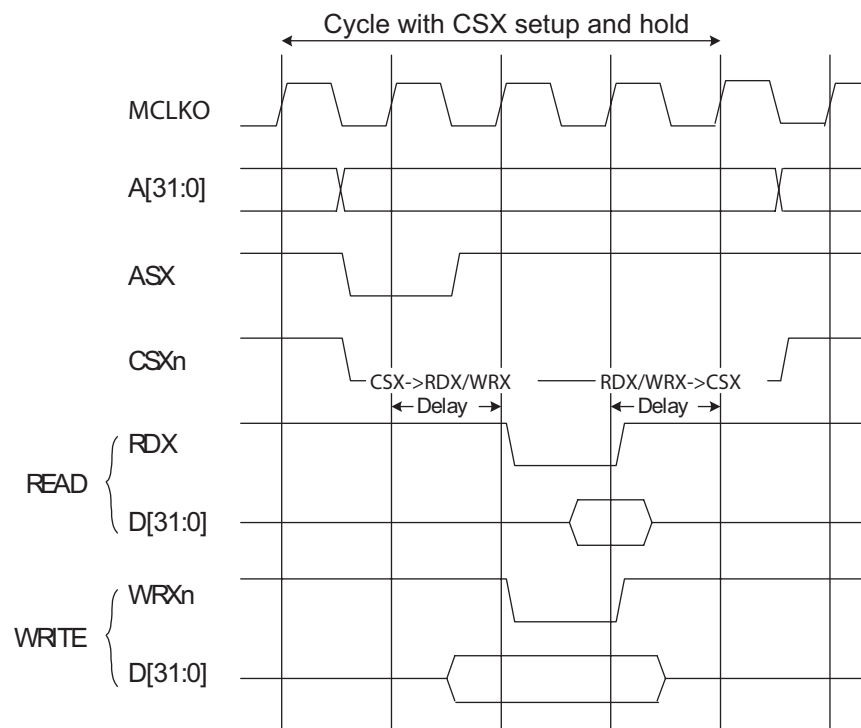
31.5.9 CSXn -> RDX/WRXn/WEX Setup and RDX/WRXn/WEX -> CSXn Hold Setting

This section shows the operation timing for the CSXn -> RDX/WRXn/WEX setup and RDX/WRXn/WEX -> CSXn hold settings.

■ Operation Timing for the CSXn -> RDX/WRXn/WEX Setup and RDX/WRXn/WEX -> CSXn Hold Settings

Figure 31.5-9 "Timing Chart for the CSXn -> RDX/WRXn/WEX Setup and RDX/WRXn/WEX -> CSXn Hold Settings" shows the operation timing for (TYP3-0=0000_B AWR=000B_H).

Figure 31.5-9 Timing Chart for the CSXn -> RDX/WRXn/WEX Setup and RDX/WRXn/WEX -> CSXn Hold Settings



- Setting 1 for the W01 bit of the AWR register enables the CSXn -> RDX/WRXn/WEX setup delay to be set. Set this bit to extend the period between chip select assertion and read/write strobe.
- Setting 1 for the W00 bit of the AWR register enables the RDX/WRXn/WEX -> CSXn hold delay to be set. Set this bit to extend the period between read/write strobe negation and chip select negation.
- The CSXn -> RDX/WRXn/WEX setup delay (W01 bit) and RDX/WRXn/WEX -> CSXn hold delay (W00 bit) can be set independently.
- When making successive accesses within the same chip select area without negating the chip select, neither a CSXn -> RDX/WRXn/WEX setup delay nor an RDX/WRXn/WEX -> CSXn hold delay is inserted.
- If a setup cycle for determining the address or a hold cycle for determining the address is needed, set 1 for the address -> CSXn delay setting (W02 bit of the AWR register).

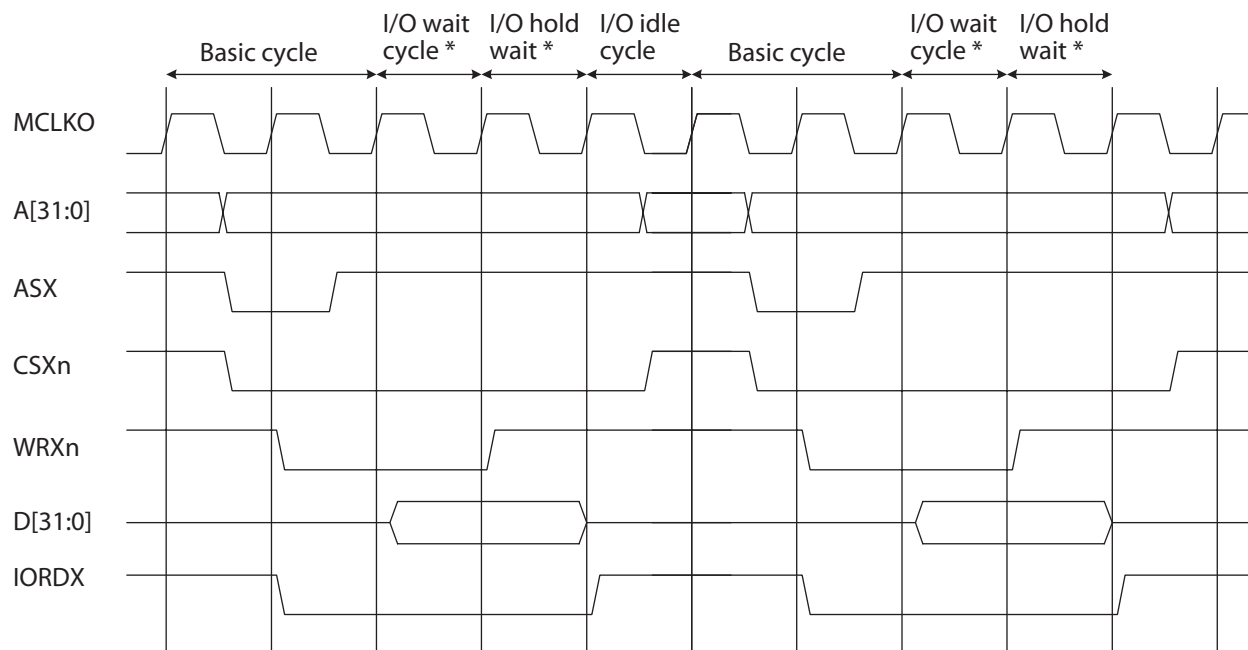
31.5.10 DMA Fly-By Transfer (I/O -> Memory)

This section shows the operation timing for DMA fly-by transfer (I/O -> memory).

■ Operation Timing for DMA Fly-By Transfer (I/O -> Memory)

Figure 31.5-10 "Timing Chart for DMA Fly-By Transfer (I/O -> Memory)" shows the operation timing for (TYP3-0=0000_B, AWR=0008_H, IOWR=51_H). This timing chart shows a case in which a wait is not set on the memory side.

Figure 31.5-10 Timing Chart for DMA Fly-By Transfer (I/O -> Memory)



- Setting 1 for the HLD bit of the IOWR0-3 registers enables the I/O read cycle to be extended by one cycle.
- Setting bits IW3-0 of the IOWR0-3 registers enables 0-15 wait cycles to be inserted.
- If wait is also set on the memory side (AWR15-12 is not 0), the larger value is used as the wait cycle after comparison with the I/O wait (IW3-0 bits).

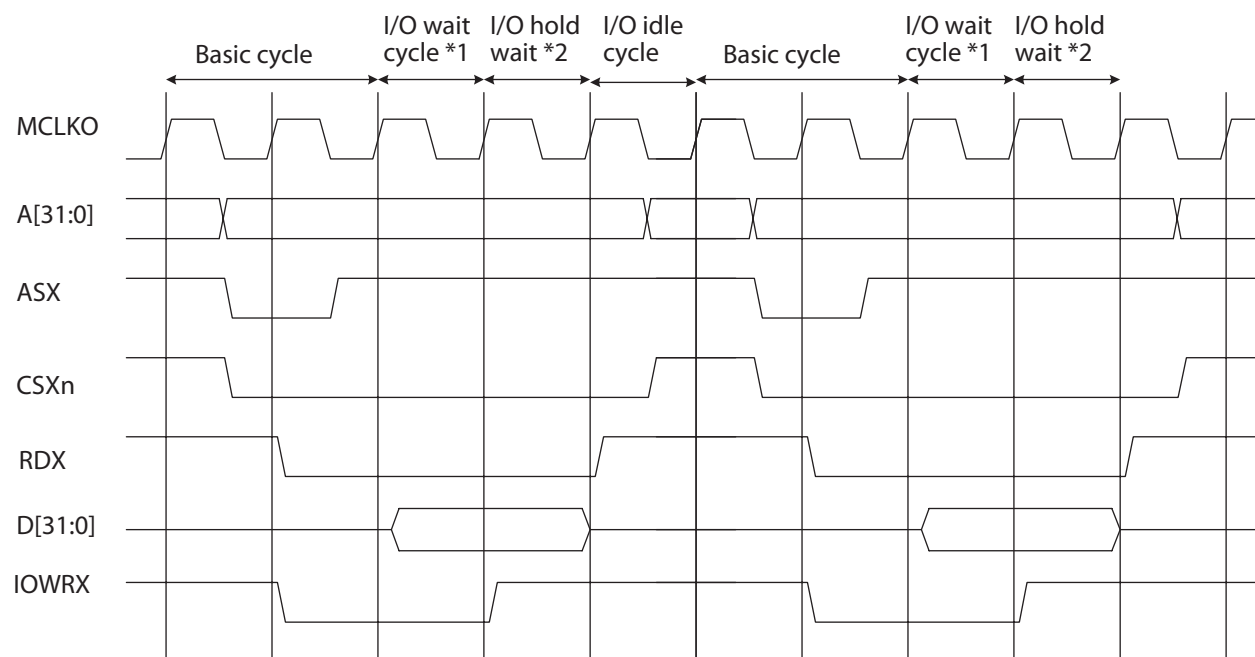
31.5.11 DMA Fly-By Transfer (Memory -> I/O)

This section shows the operation timing for DMA fly-by transfer (memory -> I/O).

■ Operation Timing for DMA Fly-By Transfer (Memory -> I/O)

Figure 31.5-11 "Timing Chart for DMA Fly-By Transfer (Memory -> I/O)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=51_H). This timing chart shows a case in which a wait is not set on the memory side.

Figure 31.5-11 Timing Chart for DMA Fly-By Transfer (Memory -> I/O)



- Setting 1 for the HLD bit of the IOWR0-3 registers enables the I/O read cycle to be extended by one cycle.
- Setting the WR1,0 bits of the IOWR0-3 registers enables 0-3 write recovery cycles to be inserted.
- If the write recovery cycle is set to 1 or more, a write recovery cycle is always inserted after write access.
- Setting bits IW3-0 of the IOWR0-3 registers enables 0-15 wait cycles to be inserted.
- If wait is also set on the memory side (AWR15-12 is not 0), the larger value is used as the wait cycle after comparison with the I/O wait (IW3-0 bits).

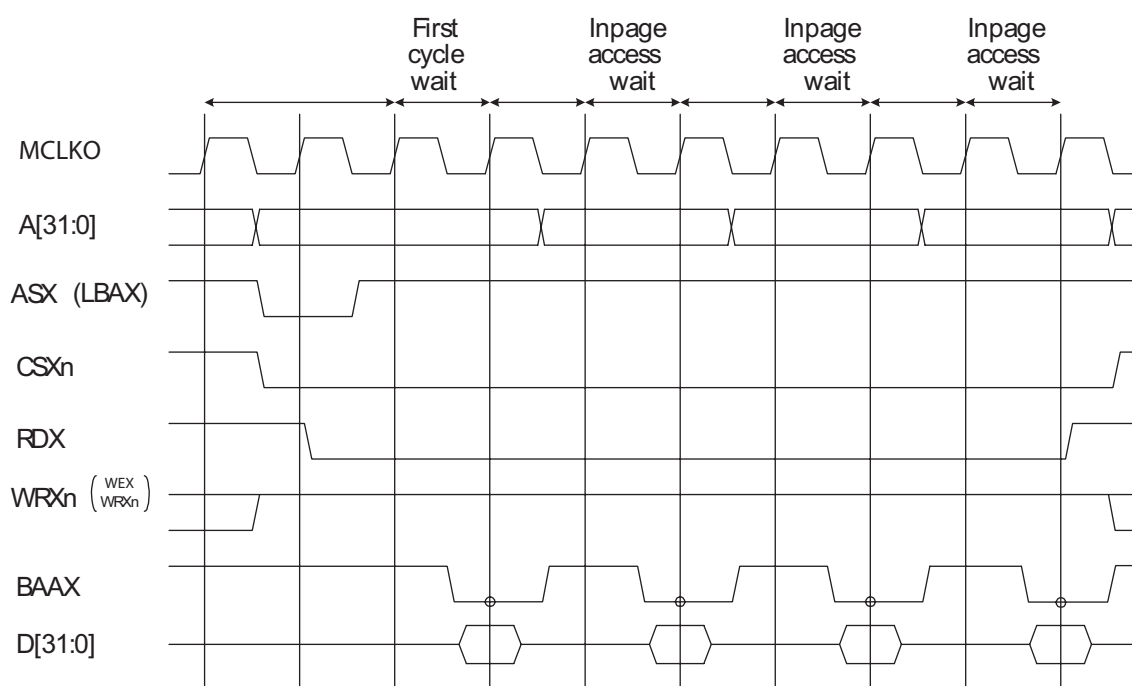
31.6. Burst Access Operation

In the external bus interface, the operation that transfers successive data items in one access sequence is called burst access. The normal access cycle (that is, not burst access) is called single access. One access sequence starts with an assertion of ASX and CSX_n and ends with negation of CSX_n. Multiple data items (two or more units of data) are transferred in sequence. This section explains burst access operation.

■ Burst Access Operation

Figure 31.6-1 "Timing chart for burst access" shows the operation timing chart for (first wait cycle=1, inpage access wait cycle=1, TYP3-0=0000_B, AWR=1108_H).

Figure 31.6-1 Timing Chart for Burst Access



- In addition to more efficient use of access cycles when a sizable amount of data of asynchronous memory such as page mode ROM and burst flash memory is read, burst cycles can also be used for reading from normal asynchronous memory.
- The access sequence when burst cycles are used can be divided into the following two types:
 - First access cycle

The first access cycle is the start cycle for the burst access and operates in the same way as the normal single access cycle.

For first read access, D31-0 is read when MCLKO rises in the cycle in which the wait cycle ended after RDX was asserted.

- Page access cycle

The page access cycle is a cycle following the first access cycle in which both CSX_n and RDX (read strobe) are asserted. Wait cycles that are different from those set for a single cycle can be set. The page access cycle is repeated while access remains in the address boundary determined by the burst length setting. When

MB91460 Series

access within the address boundary ends, burst access terminates and CSXn is negated.

For page read access, D31-0 is read when MCLKO rises in the cycle in which the wait cycle ended after address was changed.

- Setting of the W15-W12 bits of the AWR register enable the first 0-15 wait cycles to be inserted. At this point, the minimum number of the first access cycles is the wait cycles + 2 cycles (three cycles in the timing chart shown in [Figure 31.6-1](#) "Timing Chart for Burst Access").
- Setting of the W11-W08 bits of the AWR register enables 0-15 page wait cycles to be inserted. At this point, the page access cycles can be obtained from the page wait cycles + 1 cycle (Two cycles in the timing chart shown in [Figure 31.6-1](#) "Timing Chart for Burst Access")
- Setting of the BST bits of the ACR register enables the burst length to be set as 1, 2, 4, or 8. If the burst length is set to 1, single access mode is set and only the first cycle is repeated. However, if the data bus width is set to 32 bits (the DBW bits of the ACR register are 10_B), set the burst length to 4 or less (A malfunction occurs if the burst length is set to 8).
- If burst access is enabled, burst access is used when prefetch access or transfer with a larger size than the specified data bus width is performed. For example, if word access to an area whose data bus width is set to 8 bits and burst length to 4 is performed, access of 4 bursts is performed once instead of repeating byte access four times.
- Since RDY input is ignored in areas for which burst access is set, do not set TYP3-0=0xx1_B.
- The LBAX and BAAX signals are designed for burst FLASH memory. LBAX indicates the start of access and BAAX indicates the address increment.
- A31-0 is updated after the wait cycles that were set during burst access.

31.7. Address/data Multiplex Interface

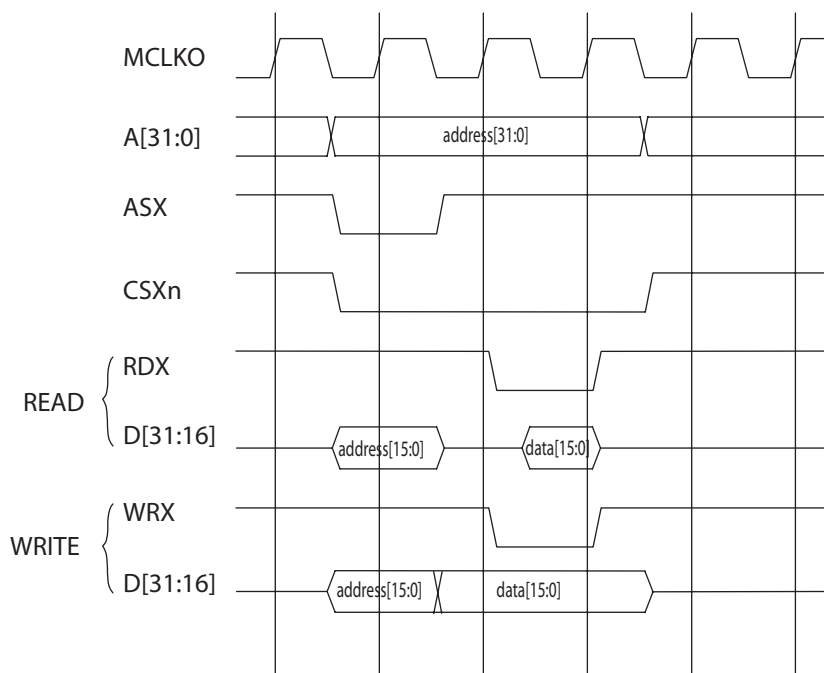
This section explains the following three cases of operation of the address/data multiplex interface:

- Without external wait
- With external wait
- CSXn -> RDX/WRXn/WEX setup

■ Without External Wait

Figure 31.7-1 "Timing Chart for the Address/Data Multiplex Interface (without External Wait)" shows the operation timing chart for (TYP3-0=0100_B, AWR=0008_H).

Figure 31.7-1 Timing Chart for the Address/Data Multiplex Interface (without External Wait)

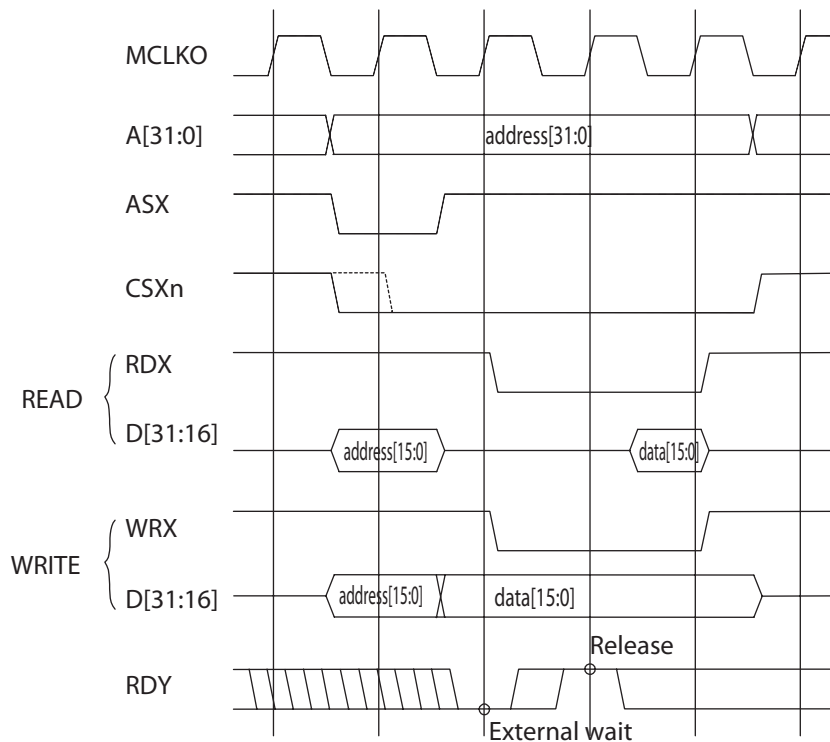


- Making a setting such as TYP3-0=01xx_B in the ACR register enables the address/data multiplex interface to be set.
- If the address/data multiplex interface is set, set 8 bits or 16 bits for the data bus width (DBW1-0 bits).
- In the address/data multiplex interface, the total of 3 cycles of 2 address output cycles + 1 data cycle becomes the basic number of access cycles.
- In the address output cycles, ASX is asserted as the output address latch signal.
- As with a normal interface, the address indicating the start of access is output to A31-0 during the time division bus cycle. Use this address if an address with more than 8/16 bits in the address/data multiplex interface should be used.
- As with the normal interface, auto-wait (AWR15-12), read -> write idle cycle (AWR7-6), write recovery (AWR5-4), address -> CSXn delay (AWR2), CSXn -> RDX/WRXn/WEX setup delay (AWR1), and RDX/WRXn/WEX -> CSXn hold delay (AWR0) can be set.
- In areas for which the address/data multiplex interface is set, set 1(DBW1-0=00_B) as the burst length.

■ With External Wait

Figure 31.7-2 "Timing Chart for the Address/Data Multiplex Interface (with External Wait)" shows the operation timing chart for (TYP3-0=0101_B, AWR=1008_H).

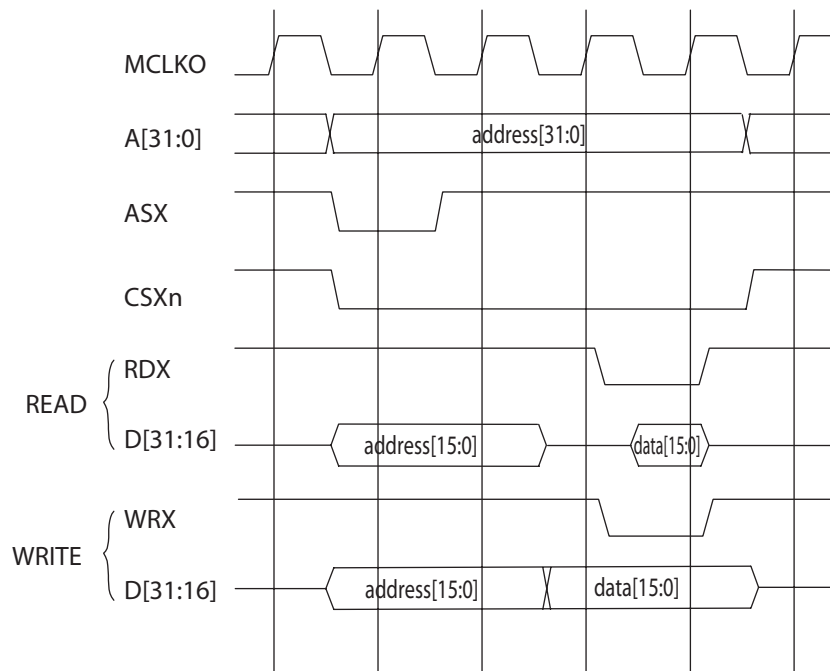
Figure 31.7-2 Timing Chart for the Address/Data Multiplex Interface (with External Wait)



Making a setting such as TYP3-0=01x1_B in the ACR register enables RDY input in the address/data multiplex interface.

■ CSXn -> RDX/WRXn/WEX Setup

Figure 31.7-3 "Timing Chart for the Address/Data Multiplex Interface (CSXn -> RDX/WRXn/WEX Setup)" shows the operation timing chart for (TYP3-0=0101_B, AWR=100B_H).

Figure 31.7-3 Timing Chart for the Address/Data Multiplex Interface ($\overline{CSXn} \rightarrow \overline{RD}/\overline{WRn}$ Setup)

Setting 1 for the $\overline{CSXn} \rightarrow \overline{RDX}/\overline{WRXn}/\overline{WEX}$ setup delay (AWR1) enables the multiplex address output cycle to be extended by one cycle as shown in Figure 31.7-3 "Timing Chart for the Address/Data Multiplex Interface ($\overline{CSXn} \rightarrow \overline{RDX}/\overline{WRXn}/\overline{WEX}$ Setup)", allowing the address to be latched directly to the rising edge of ASX. In this setting the ASX is used as an ALE (Address Latch Enable) strobe without using MCLKO.

MB91460 Series

31.8. Prefetch Operation

This section explains the prefetch operation.

■ Prefetch Operation

The external bus interface controller contains a prefetch buffer consisting of 16 x 8 bits.

If the PSUS bit of the TCR register is 0 and read access to an area to which the PFEN bit of the ACR register is set to 1 occurs, the subsequent address is prefetched and then stored in the prefetch buffer.

If the stored address is accessed from the internal bus, the lookahead data in the prefetch buffer is returned without external access being performed. This can reduce the wait time for successive accesses to the external bus areas.

● Basic conditions for starting external access using prefetch

External bus access using prefetch occurs when the following conditions are met:

- The PSUS bit of the TCR register is 0.
- Neither SLEEP state nor STOP state is set.
- Read access by the external bus to a chip select area for which prefetch is enabled has been performed. DMA access and read access by a read modified write system instruction, however, are excluded.
- No external bus access request (external bus area access to an area for which prefetch is not enabled or DMA transfer with an external bus area) other than the prefetch access has occurred.
- The part of the prefetch buffer for the next operation of capturing the prefetch access is completely empty.

While the above conditions are met, the prefetch access will continue. If external bus area access to an area for which prefetch is not enabled occurs after prefetch access, prefetch access to the area for which prefetch is enabled will continue as long as the prefetch buffer clear conditions are not met.

For an access that mixes multiple prefetch-enabled areas and multiple prefetch-disabled areas, the prefetch buffer always holds data of the prefetch-enabled area accessed last. Since, in this case, access to prefetch-disabled areas does not affect the prefetch buffer state at all, data in the prefetch buffer is not wasted even if prefetch-disabled data access and prefetch-enabled instruction fetch are mixed.

● Optional clear for temporary stopping of a prefetch access

Setting 1 for the PSUS bit of the TCR register temporarily stops a prefetch. The prefetch can be restarted by setting the PSUS bit to 0. At this point, the contents of the buffer are retained if no error occurs or a buffer clear such as occurs when the PCLR bit is set does not occur.

Setting 1 for the PCLR bit of the TCR register completely clears the prefetch buffer. Clear the buffer by setting the PSUS bit when prefetch is interrupted.

Prefetch is temporarily stopped for the minimum unit (64 KB) of the boundary=chip select area where the high-order 16 bits of an address change. If the boundary is crossed, first a buffer read error occurs and then prefetch starts in a new area.

● Unit for one prefetch access operation

The unit for one prefetch access operation is determined by the DBW bits (bus width) and BST bits (burst length).

Prefetch access always occurs with the full size of the bus width specified by the DBW bits and access for the count of the burst length set by the BST bits in one access operation is performed. That is, if any value other than 00_B is set for the BST bits, the prefetch always occurs in page mode/burst mode. Keep in mind whether ROM/RAM is conformable and enough access time is applicable. (Set an appropriate value bits W15-08 bits of the AWR register).

During burst access, successive accesses occur only within the address boundary that is determined by the burst length. Thus, if the boundary is crossed, for example, 4 bytes of free space are available in the buffer, these 4 bytes cannot be accessed in one operation (If the prefetch buffer starts at xxxxxx0E_H, 4 bytes of free space are available in the buffer, and two bursts are set even though the bus width is 16 bits, only 2 bytes, xxxxxx0E_H and xxxxxx0F_H, can be captured in the next prefetch access).

The following provides two examples:

- Area whose bus width is set to 16 bits and whose burst length is set to 2

The amount of data read into the buffer in one prefetch operation is 4 bytes. In this case, prefetch access is delayed until 4 bytes of free space are available in the prefetch buffer.

- Area whose bus width is set to 8 bits and whose burst length is set to 8

The amount of data read into the buffer in one prefetch operation is 8 bytes. In this case, prefetch access is delayed until 8 bytes of free space are available in the prefetch buffer.

● Burst length setting and prefetch efficiency

If requests for external bus access, other than prefetch access, to or errors in the prefetch buffer occur during one operation of prefetch access as explained in the previous bullet, "Unit of one prefetch access operation," these access requests must wait until access to the prefetch buffer that is being executed is completed.

Thus, if the burst length is too long, the efficiency and reaction of bus access other than prefetch may be degraded. If, on the other hand, the burst length is set to 1, many read cycles may be wasted even if burst/page access memory is connected because single access is always performed.

If settings are made so that the amount of data read in one prefetch access operation is large, prefetch access can be started only after free space in the prefetch buffer for this amount is available. Thus, access to the prefetch buffer is infrequent, and the external bus tends to be idle. For example, if the bus width is set to 16 bits and the burst length is set to 8, the amount of data read into the buffer in one prefetch operation is 16 bytes. Thus, a new prefetch access can be started only after the prefetch buffer is completely empty.

Adjust the optimum burst length to suit use and the environment after taking the above into consideration. Generally, when connecting asynchronous memory to which burst/page access cannot be applied, it is best to set the burst length to 1 (single access). Conversely, when memory whose burst/page access cycle is short is connected, it is better to set the burst length to any value other than 1 (single access). In this case, it is best to make the setting so that 8 bytes (half of the buffer) are read in one read operation according to the bus width. However, the optimum condition varies with the frequency of external access and varies with the frequency divide-by rate setting of the external access clock.

● Reading from the prefetch buffer

Data stored in the prefetch buffer is read in response to access from the internal bus if an address matches, and no external access is performed. In reading from the buffer, addresses can be hit (up to 16 bytes) if they are in the forward direction but not continuous, so that a second read from the external bus is avoided, if possible, even for a short forward branch.

If the address currently being accessed for prefetch matches during access from the internal bus, a wait signal is returned internally before data is captured after prefetch access is completed. In this case, no buffer error occurs.

If an address in the prefetch buffer matches when a read is performed for DMA transfer, data in the prefetch buffer is not used, and instead, external data is read by the external bus. In this case, a buffer error occurs. The prefetch is not continued and no prefetch access is performed until a new external access operation to a prefetch-enabled area occurs.

● Clearing/updating the prefetch buffer

If either of the following conditions is met, the prefetch buffer is completely cleared:

- If 1 is written to the PCLR bit of the TCR register

MB91460 Series

- If a buffer read error occurs. A buffer read error is if any of the following events occurs:
 - When no address is found in the buffer that matches in an to read from a prefetch-enabled area. In this case, the external bus is accessed again. Data read in this case is not stored in the buffer, but the prefetch access is started from the subsequent address to store addresses in the buffer.
 - In an access to read from a prefetch-enabled area with a read modified system instruction. In this case, the external bus is accessed again. Data read in this case is not stored in the buffer. Also, no prefetch access is performed (This is because data is written to the next address).
 - In an access to read from a prefetch-enabled area for DMA transfer. In this case, the external bus is accessed again. Data read in this case is not stored in the buffer. Also, no prefetch access is performed.
- If a buffer write hit occurs. A buffer write hit is as follows:
 - When the address of just one byte that matches is found in the buffer in an access to write to a prefetch-enabled area. In this case, the external bus is accessed again, but no prefetch access is performed before a new read access occurs.

Only part of the prefetch buffer is cleared when the following condition is met:

- If a buffer read hit occurs

In this case, only the part of the buffer before the hit address is cleared.

● **Restrictions on prefetch-enabled areas**

If prefetch to a little endian area is enabled, be sure to access the area using word access. If data read into the prefetch buffer is accessed with any length other than word length, the correct endian conversion is not performed and thus the wrong data will be read. This is due to hardware restrictions related to the endian conversion mechanism.

31.9. SDRAM/FCRAM Interface Operation

This section describes the operations of the SDRAM/FCRAM interface.

■ SDRAM/FCRAM interface

The chip select areas can be used as SDRAM/FCRAM interface by setting the TYP3 to TYP0 bits in the area configuration register (ACR) to 100X_B.

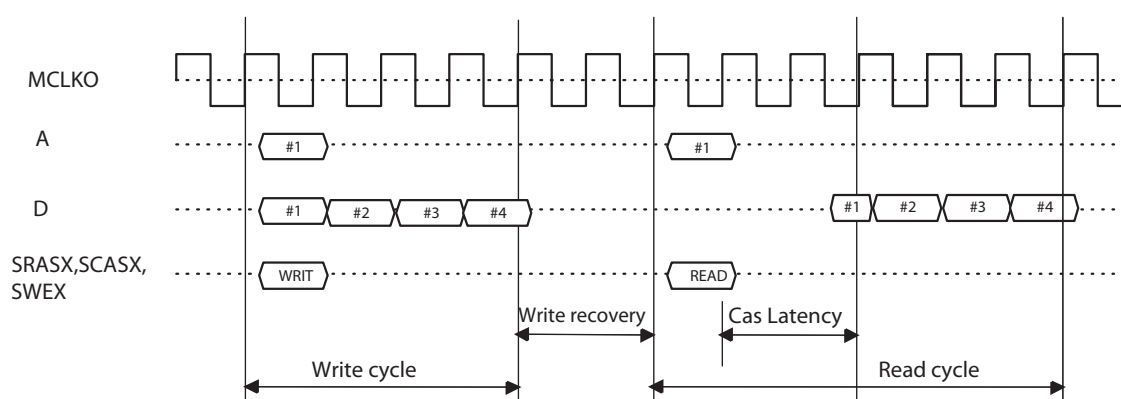
This section provides timing charts to describe the following operations of the SDRAM/FCRAM interface.

- Burst read/write (Settings: Page hit, CAS latency 2)
- Single read/write (Settings: Page hit, CAS latency 3, auto - precharge OFF)
- Single read (Settings: Page miss, CAS latency 3, auto - precharge OFF)
- Single read/write (Settings: CAS latency 1, TYP 1001_B, auto - precharge ON)
- Auto - refresh

■ Burst Read/Write Operation Timing

Figure 31.9-1 shows the operation timings assuming that page hits and CAS latency 2 are set.

Figure 31.9-1 Burst Read/Write Timing Chart

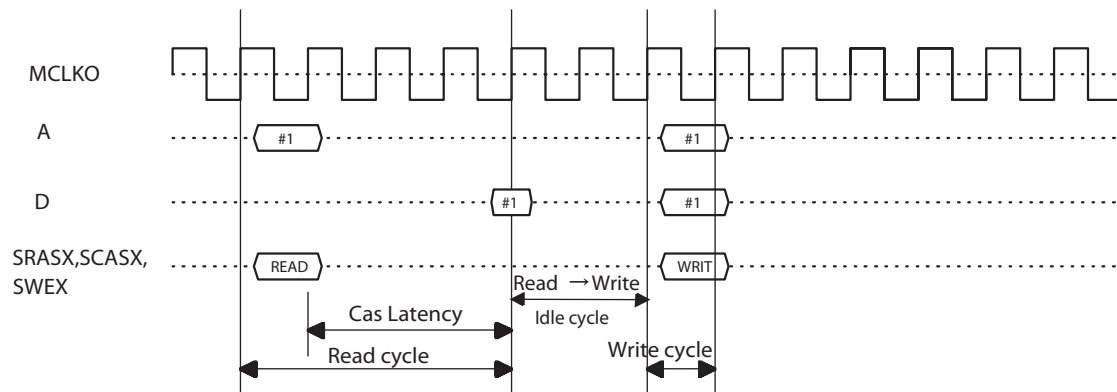


- All of the A13 to A0 pins may not be used depending on the SDRAM capacity. See Section "Memory Connection Examples".
- The MCLKO is a clock signal input to SDRAM. Signals such as addresses, data, and commands are input to SDRAM at the rise of the MCLKO.
- Set the W05 and W04 bits in the area wait register (AWR) to the write recovery cycle according to the SDRAM/FCRAM standards.
- Set the W10 to W08 bits in the area wait register (AWR) to the CAS latency according to the SDRAM/FCRAM standards.
- Set the burst length using the BST bit in the area configuration register (ACR).

■ Single Read/Write Operation Timing

Figure 31.9-2 shows the operation timings assuming that page hits, CAS latency 3, and no auto - precharge are set.

Figure 31.9-2 Single Read/Write Timing Chart

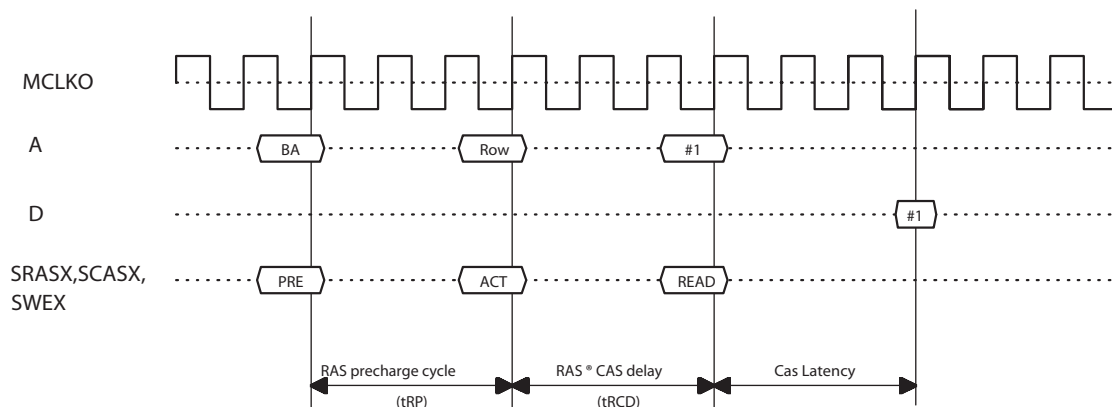


Set the W07 and W06 bits in the area wait register (AWR) to the read - to - write idle cycle according to the SDRAM/FCRAM standards.

■ Single Read Operation Timing

Figure 31.9-3 shows the operation timings assuming that page misses, CAS latency 3, and no auto - precharge are set.

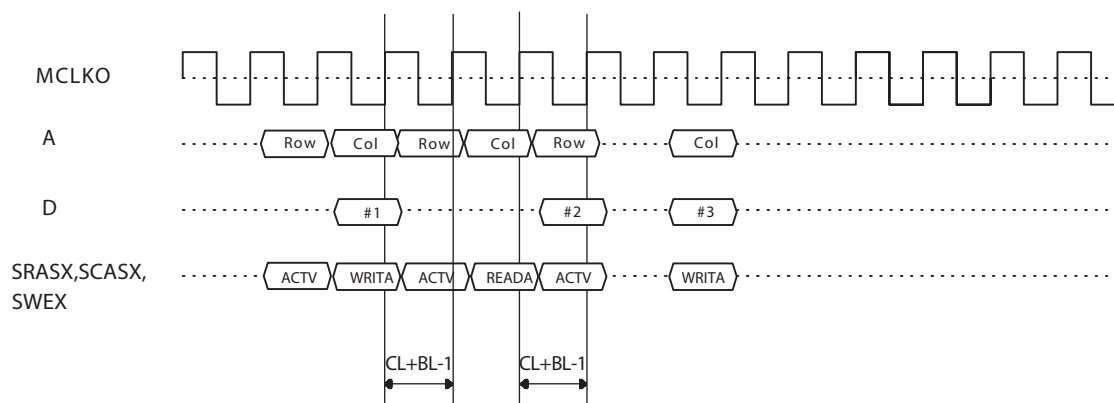
Figure 31.9-3 Single Read Timing Chart



- When a page miss occurs, a read operation is performed after the PRE charge and ACTV commands are issued.
- Set the W01 and W00 bits in the area wait register (AWR) to the RAS precharge cycle (tRP) according to the SDRAM/FCRAM standards.
- Set the W14 to W12 bits in the area wait register (AWR) to the RAS - to - CAS delay (tRCD) according to the SDRAM/FCRAM standards.

■ Single Read/Write Operation Timing

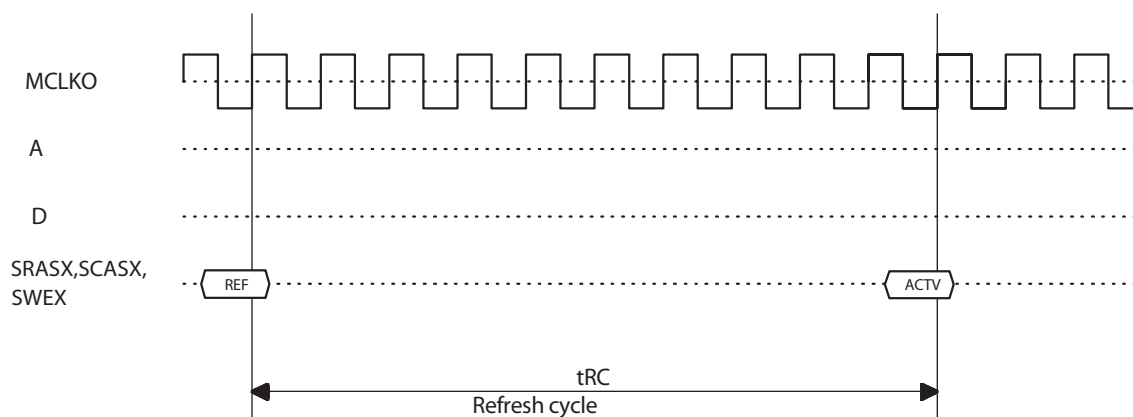
Figure 31.9-4 shows the operation timings assuming that CAS latency 1, TYP = 1001_B, and auto - precharge are set.

Figure 31.9-4 Single Read/Write Timing Chart

- Setting TYP to 1001_B causes a read/write command with auto - precharge to be issued. Since the cycle from READA/WRITA issuance to ACTV issuance is fixed at CL + BL - 1, however, TYP can be set to 1001_B only when FCRAM is connected.
- This timing is effective, for example, for recurring page misses as it eliminates the cycle for issuing the PRE command.

■ Auto - refresh Operation Timing

Figure 31.9-5 shows auto - refresh operation timings.

Figure 31.9-5 Auto - refresh Timing Chart

- The refresh command is issued every " refresh control register's (RCR's) RFINT5 - RFINT0 value x 32 " cycles and access is restarted upon completion of each refresh.
- Set the TRC bit in the refresh control register (RCR) according to the SDRAM/FCRAM standards.
- Satisfy the maximum RAS active time as well.

31.9.1 Self Refresh

This section describes self - refreshing.

■ Self Refresh

Writing 1 to the SELF bit in the refresh control register (RCR) causes the SDRAM/FCRAM interface to initiate the self - refresh transition sequence.

After executing auto - refreshing the number of times set in the RFC2 to RFC0 bits, the SDRAM/FCRAM interface issues the SELF command to SDRAM/FCRAM to enter the self - refresh mode.

The device is released from the self - refresh mode either when 0 is written to the SELF bit or read/write access to SDRAM/FCRAM occurs.

The SDRAM/FCRAM interface issues the SELFX command to execute auto - refreshing the number of times set in the RFC2 to RFC0 bits upon detection of writing 0 to the SELF bit or access to SDRAM/FCRAM in the self - refresh mode.

Even when access to SDRAM/FCRAM by DMA transfer occurs after setting the self - refresh mode and putting the chip into SLEEP state, the self - refresh mode is canceled.

- Self - refresh mode transition procedure

1. Set SELF bit to "1".
2. Issue the REF command the number of times set in the RFC2 to RFC0 bits.
3. Issue SELF command

- Self - refresh mode reset procedure

1. Set the SELF bit to 0 or access to SDRAM/FCRAM.
2. Issue SELFX command
3. Issue the REF command the number of times set in the RFC2 to RFC0 bits.
4. Transition to the normal access state

31.9.2 Power-on Sequence

This section describes the power - on sequence.

■ Power-on Sequence

Setting the PON bit in the refresh control register (RCR) to 1 initiates the power - on sequence.

Take the following steps to set the PON bit to 1 for transition to the power - on sequence.

1. Reserve the clock stabilization wait time specified in the SDRAM/FCRAM manual.
2. Set ACR, AWR, MCRA(B).
3. Set the CSER to enable the area to which SDRAM/FCRAM has been connected.
4. Set the PON bit to 1 while setting the RCR value.

Taking the above steps causes the SDRAM/FCRAM interface to execute the following power - on sequence.

5. Execute the PALL command.
6. Execute the REF command eight times.
7. The mode register is set according to the BST bit in the ACR, CL (CAS Latency) bit in the AWR, and the WBST bit in the MCRA.
8. Transition to the normal access state

31.9.3 Connecting SDRAM/FCRAM to Many Areas

This section shows the connecting SDRAM/FCRAM to many areas.

■ Connecting SDRAM/FCRAM to Many Areas

SDRAM/FCRAM can basically be set for all chip select areas. When connecting SDRAM/FCRAM to several areas, connect the same type of modules. Also it needs considerations about bus load when connecting SDRAM/FCRAM to several areas.

More precisely, connect the modules common in the following register settings.

- Area configuration register (ACR): Set all of the DBW1 - DBW0, BST1 - BST0, and TYP3 - TYP0 bits to the same.
- Area wait register (AWR): Set all the bits to the same.
- Memory setting register (MCR): All the settings are the same as the registers are common.)
- Refresh control register (RCR): All the settings are the same as the registers are common.)

To enable the two areas at a time, execute the power - on sequence, auto - refresh, and self - refresh at the same time.

31.9.4 Address Multiplexing Format

This section describes the address multiplexing format.

■ Address Multiplexing Format

SDRAM/FCRAM access addresses correspond to row, bank, and column addresses differently depending on the settings of the ASZ3 to ASZ0, DBW1 and DBW0, PSZ2 to PSZ0, and BANK bits.

Addresses are arranged in the order of Column, BANK, and Row addresses, starting from the least significant bit.

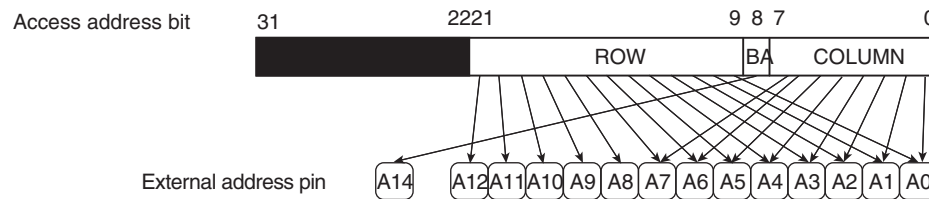
Set each bit as shown below.

- ASZ3 to ASZ0 bits: Set these bits to the total amount of SDRAM/FCRAM connected to the corresponding area. For using two modules in parallel, set the total amount. Affects the number of row addresses.
- DBW1 and DBW0 bits: Set these bits to the data bus width. (Set the bits to " 16 bits " for connecting a pair of eight - bit modules in parallel.) Column addresses are shifted according to the data bus width setting. 8 bits: Do not shift. 16 bits: Shift one bit. 32 bits: Shift two bits.
- PSZ2 to PSZ0 bits: Set these bits to the number of column addresses used for SDRAM/FCRAM.
- BANK bit: Set this bit to the number of SDRAM/FCRAM bank addresses.

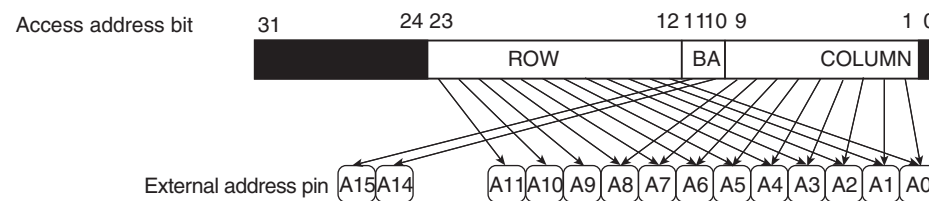
Figure 31.9-6 shows examples of combinations of access addresses and Row/BANK/Column addresses.

Figure 31.9-6 Examples of combinations of access addresses and Row/BANK/Column addresses

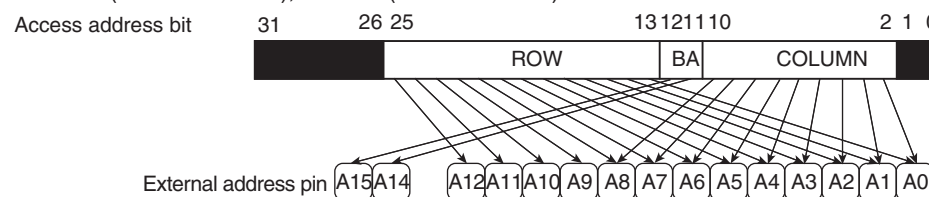
- 4M bytes (set ASZ to 0110B), 8-bit bus width (set DBW to 00B)
256 column (set PSZ to 000B), 2 banks (set BANK to 0B)



- 16M bytes (set ASZ to 1000B), 16-bit bus width (set DBW to 01B)
512 column (set PSZ to 001B), 4 banks (set BANK to 1B)



- 64M bytes (set ASZ to 1010B), 32-bit bus width (set DBW to 10B)
512 column (set PSZ to 001B), 4 banks (set BANK to 1B)



31.9.5 Memory Connection Example

This section shows the memory connection example.

Memory Connection Example

The SDRAM/FCRAM interface is connected to SDRAM/FCRAM as shown in [Table 31.9-1](#) in principle.

Table 31.9-1 SDRAM/FCRAM Interface to SDRAM/FCRAM Connection Table

SDRAM/ FCRAM interface pin	SDRAM/ FCRAM pin	Remarks
MCLKO	CLK	
MCLKE	CKE	
SRASX (ASX)	RASX	
SCASX (BAAX)	CASX	
SWEX (WEX)	WEX	

Table 31.9-1 SDRAM/FCRAM Interface to SDRAM/FCRAM Connection Table

SDRAM/ FCRAM interface pin	SDRAM/ FCRAM pin	Remarks
CSX0 to CSX7	CSX	All chip select areas can be set as SDRAM/FCRAM space.
A0 to A9	A0 to A9	Addresses do not have to be shifted depending on the bus width.
A10/AP	A10/AP	A10 for row address output; otherwise AP
A11 to A13	A11 to A13	Connected to the address used for SDRAM/FCRAM.
A14	BA0	BA for 2 bank product
A15	BA1	The pin is not used for a two - bank module.
D31 to D0	DQ	The connection changes depending on the endian method and data bus width. For detailed connection, see Section " 31.4. Endian and Bus Access " .
DQMUU, DQMUL, DQMLU, DQMLL	DQM	The connection changes depending on the endian method and data bus width. For detailed connection, see Section " 31.4. Endian and Bus Access " .

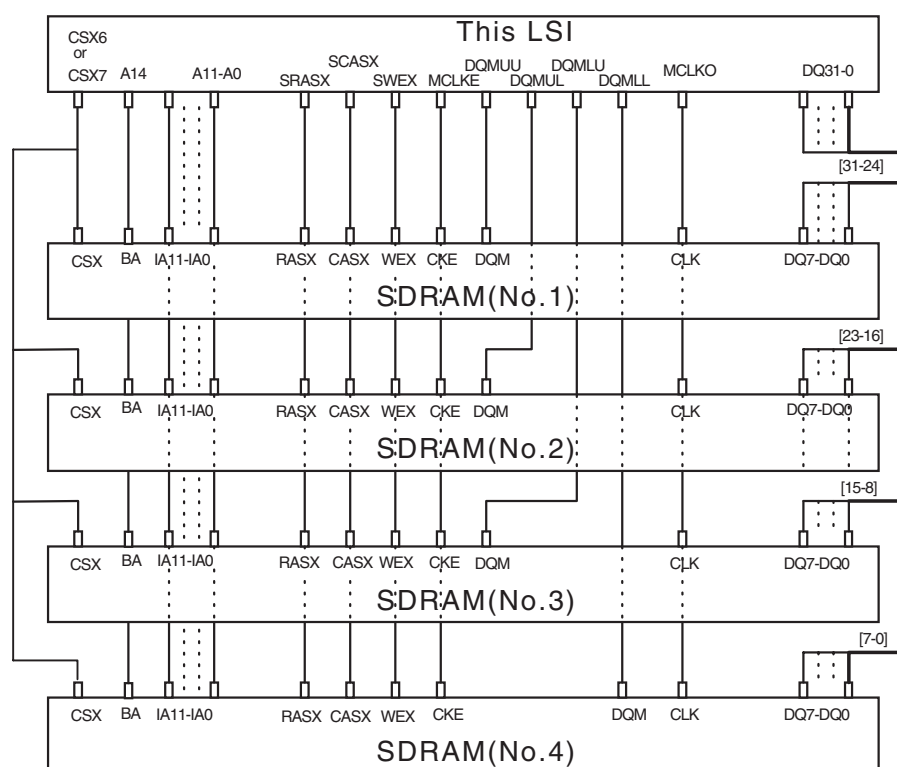
● Using 8 - bit SDRAM/FCRAM (Big endian)

Total data bus width of 32 bits: Use four SDRAM/FCRAM modules.

Total data bus width of 16 bits: Use two SDRAM/FCRAM modules.

Figure 31.9-7 shows how to use 64 - Mbit SDRAM (one bank address and 12 row addresses).

Figure 31.9-7 Using 64 - Mbit SDRAM



When SDRAM modules are used with a total data width of 16 bits, SDRAMs No. 3 and No. 4 are not required and DQ15 to DQ0 must be left open.

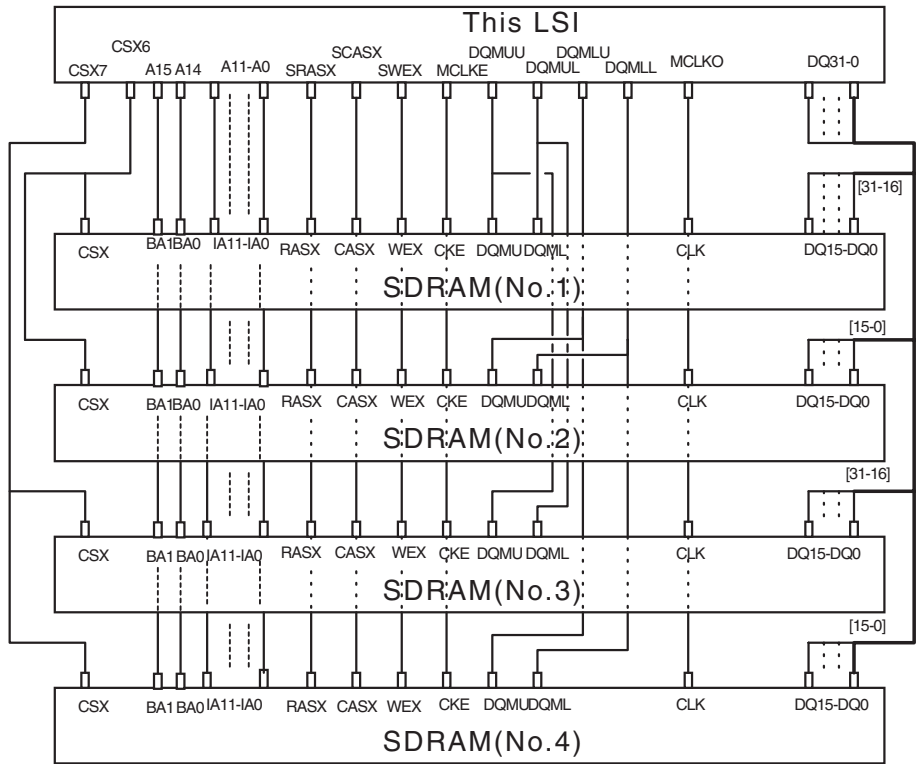
● Using 16 - bit SDRAM/FCRAM

Total data width of 32 bits: Use two or four SDRAM modules.

Total data width of 16 bits: Use one or two SDRAM modules.

Figure 31.9-8 shows how to use 64-Mbit SDRAM (two bank addresses and 12 row addresses).

Figure 31.9-8 Using 64 - Mbit SDRAM



When using one SDRAM module with a data width of 16 bits, SDRAMs No. 2, No. 3, and No. 4 are not required and DQ15 to DQ0 must be left open.

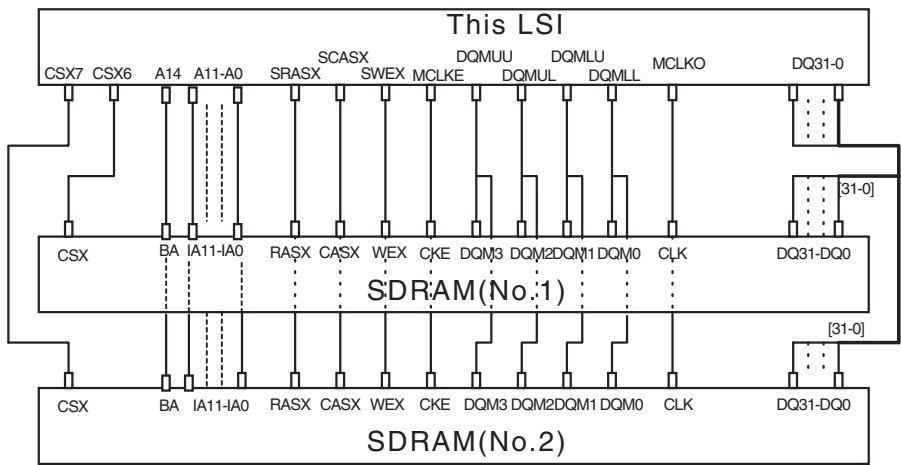
When two SDRAM modules are used with a data width of 16 bits, SDRAMs No. 2 and No. 4 are not required. When two SDRAM modules are used with a data width of 32 bits, SDRAMs No. 3 and No. 4 are not required.

● Using 32 - bit SDRAM

When the data width is 32 bits: Use one or two SDRAM modules.

Figure 31.9-9 shows 64-Mbit SDRAM (one bank address and 12 row addresses).

Figure 31.9-9 Using 64 - Mbit



SDRAM No. 2 is not required when the device is used with only one SDRAM module.

MB91460 Series

31.10. DMA Access Operation

This section explains DMA access operation.

■ DMA Access Operation

This section explains the following five DMA operations:

- DMA fly-by transfer (I/O -> memory)
- DMA fly-by transfer (memory -> I/O)
- 2-cycle transfer (internal RAM -> I/O, RAM)
- 2-cycle transfer (external -> I/O)
- 2-cycle transfer (I/O -> external)

31.10.1 DMA Fly-By Transfer (I/O -> Memory)

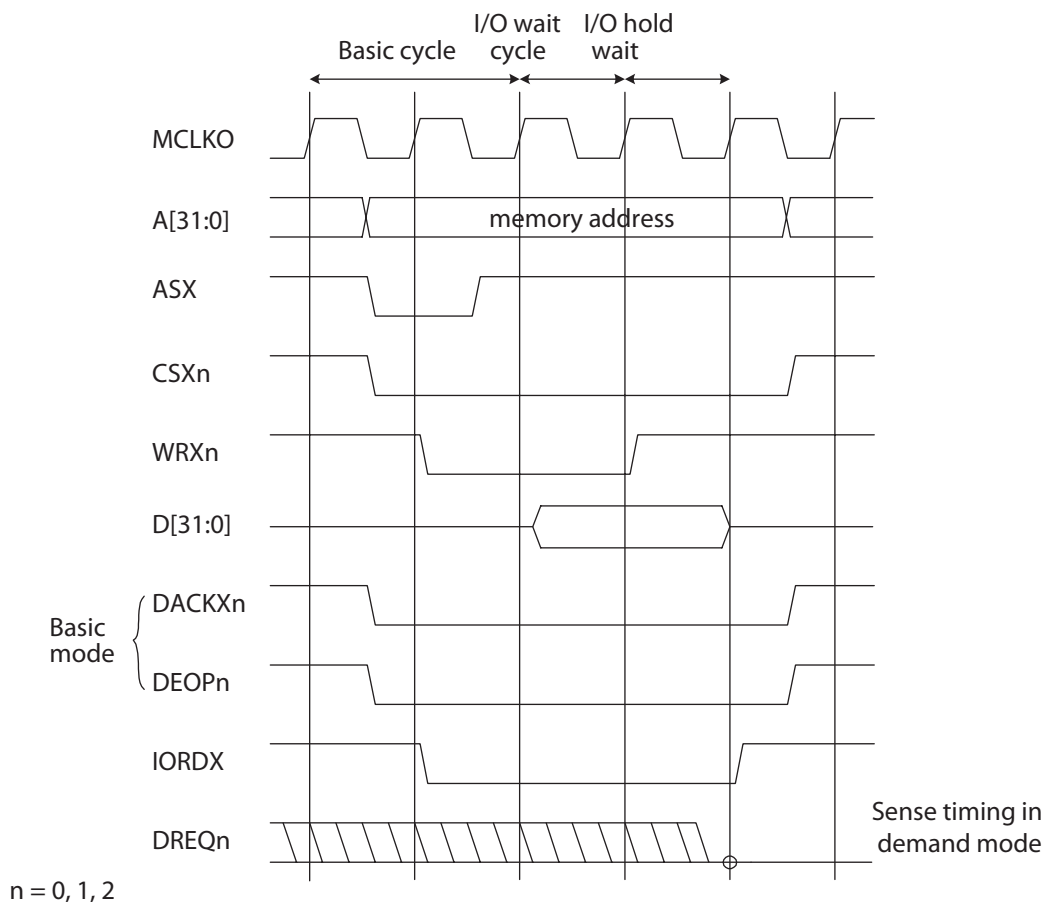
This section explains DMA fly-by transfer (I/O -> memory).

■ DMA Fly-By Transfer (I/O -> Memory)

Figure 31.10-1 "Timing Chart for DMA Fly-By Transfer (I/O -> Memory)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=41_H).

Figure 31.10-1 shows a case when a wait is not set on the memory side.

Figure 31.10-1 Timing Chart for DMA Fly-By Transfer (I/O -> Memory)



- Setting 1 for the W01 bit of the AWR register enables the CSXn -> RDX/WRXn/WEX setup delay to be set. Set this bit to extend the period between assertion of chip select and the read/write strobe.
- Setting 1 for the W00 bit of the AWR register enables the RDX/WRXn/WEX -> CSXn hold delay to be set. Set this bit to extend the period between negation of the read/write strobe and negation of chip select.
- The CSXn -> RDX/WRXn/WEX setup delay (W01 bit) and RDX/WRXn/WEX -> CSXn hold delay (W00 bit) can be set independently.
- When successive accesses are made within the same chip select area without negating the chip select, neither CSXn -> RDX/WRXn/WEX setup delay nor RDX/WRXn/WEX -> CSXn hold delay is inserted.
- If a setup cycle for determining the address or a hold cycle for determining the address is needed, set 1 for the address -> CSXn delay setting (W02 bit of the AWR register).

For I/O on the data output side, a read strobe of three bus cycles extended by the I/O wait cycle and I/O hold wait cycle is generated. For memory on the receiving side, a write strobe of two bus cycles extended by the I/O wait cycle is generated. The I/O hold wait cycle does not affect the write strobe. However, the address and CSX signal are retained until the fly-by bus access cycles end.

31.10.2 DMA Fly-By Transfer (Memory -> I/O)

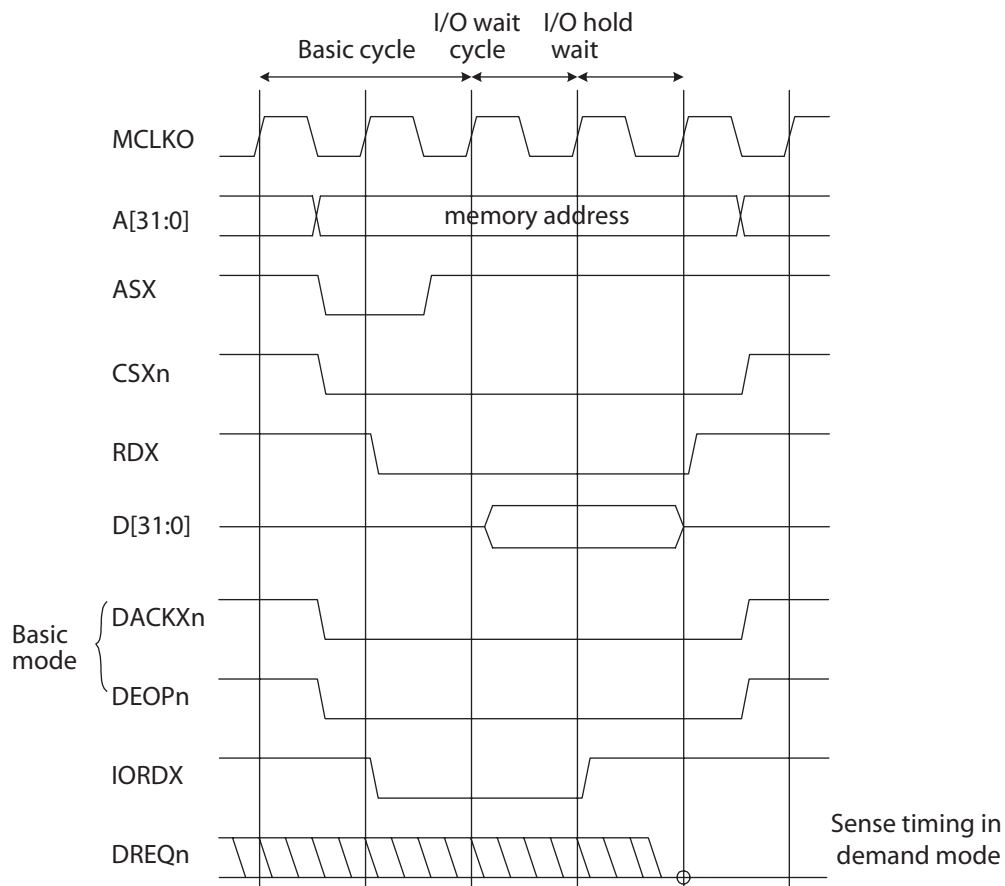
This section explains DMA fly-by transfer (memory -> I/O).

■ DMA Fly-By Transfer (Memory -> I/O)

Figure 31.10-2 "Timing chart for DMA Fly-By Transfer (Memory -> I/O)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=41_H).

Figure 31.10-2 shows a case in which a wait is not set on the memory side.

Figure 31.10-2 Timing chart for DMA Fly-By Transfer (Memory -> I/O)



- Setting 1 for the HLD bit of the IOWR0-3 registers extends the I/O read cycle by one cycle.
- Setting bits WR1-0 bits of the IOWR0-3 registers enables 0-3 write recovery cycles to be inserted.
- If the write recovery cycle is set to 1 or more, a write recovery cycle is always inserted after write access.
- Setting bits IW3-0 of the IOWR0-3 registers enables 0-15 wait cycles to be inserted.
- If wait is also set on the memory side (AWR15-12 is not 0), the larger value is used as the wait cycle after comparison with the I/O wait (IW3-0 bits).

Reference:

For memory on the data output side, a read strobe of three bus cycles extended by the I/O wait cycle and I/O hold wait cycle is generated. For I/O on the receiving side, a write strobe of two bus cycles extended by the I/O wait cycle is generated. The I/O hold wait cycle does not affect the write strobe. However, the address and

CSX signal are retained until the fly-by bus access cycles end.

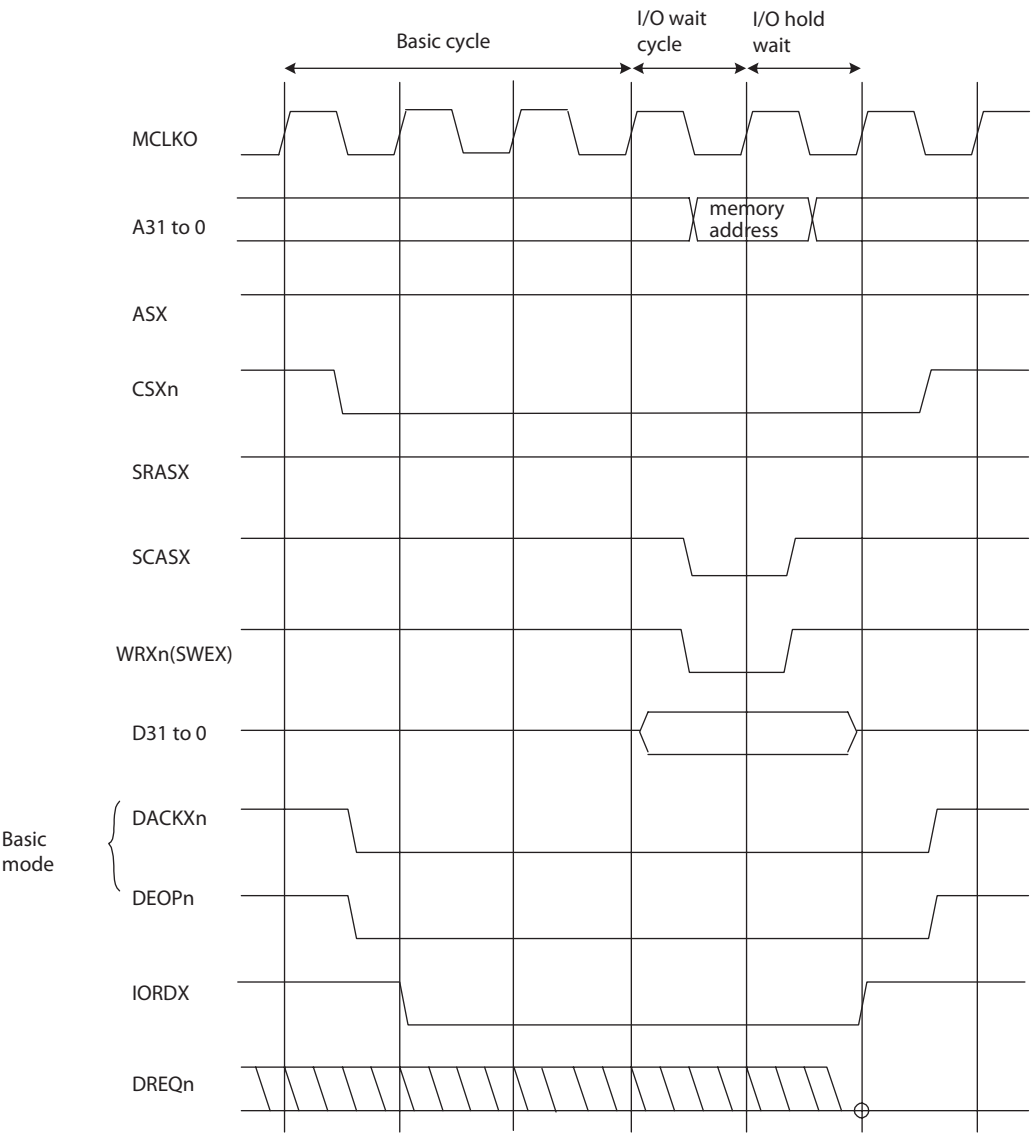
31.10.3 DMA Fly-By Transfer (I/O -> SDRAM/FCRAM)

This section describes the operation of DMA fly - by transfer (I/O device to SDRAM/FCRAM).

■ DMA Fly-By Transfer (I/O -> SDRAM/FCRAM)

Figure 31.10-3 shows an operation timing chart assuming TYP3 to TYP0 set to 1000B, AWR set to 0051H, and IOWR set to 41H.

Figure 31.10-3 Timing Chart for DMA Fly - by Transfer (I/O to SDRAM/FCRAM)



- For the I/O device on the data output side, a read strobe of three bus cycles extended by the I/O wait cycle and I/O hold wait cycle is generated.
- For SDRAM/FCRAM on the receiving side, a WRIT command is issued at the timing that allows writing after the I/O wait cycle. The I/O wait cycle may be longer depending on the SDRAM/FCRAM bank active state and SDRAM/FCRAM wait setting.
- The I/O hold wait cycle does not affect the write strobe. Note, however, that the CSX signal is retained until the fly - by bus access cycles end.
- For fly - by transfer from an I/O device to SDRAM/FCRAM, be sure to set the HLD bit in the DMAC I/O wait register (IOWR) to 1 to enable the I/O hold wait cycle.
- Fly - by transfer must always be performed between data buses having the same bus width.

31.10.4 DMA Fly-By Transfer (SDRAM/FCRAM -> I/O)

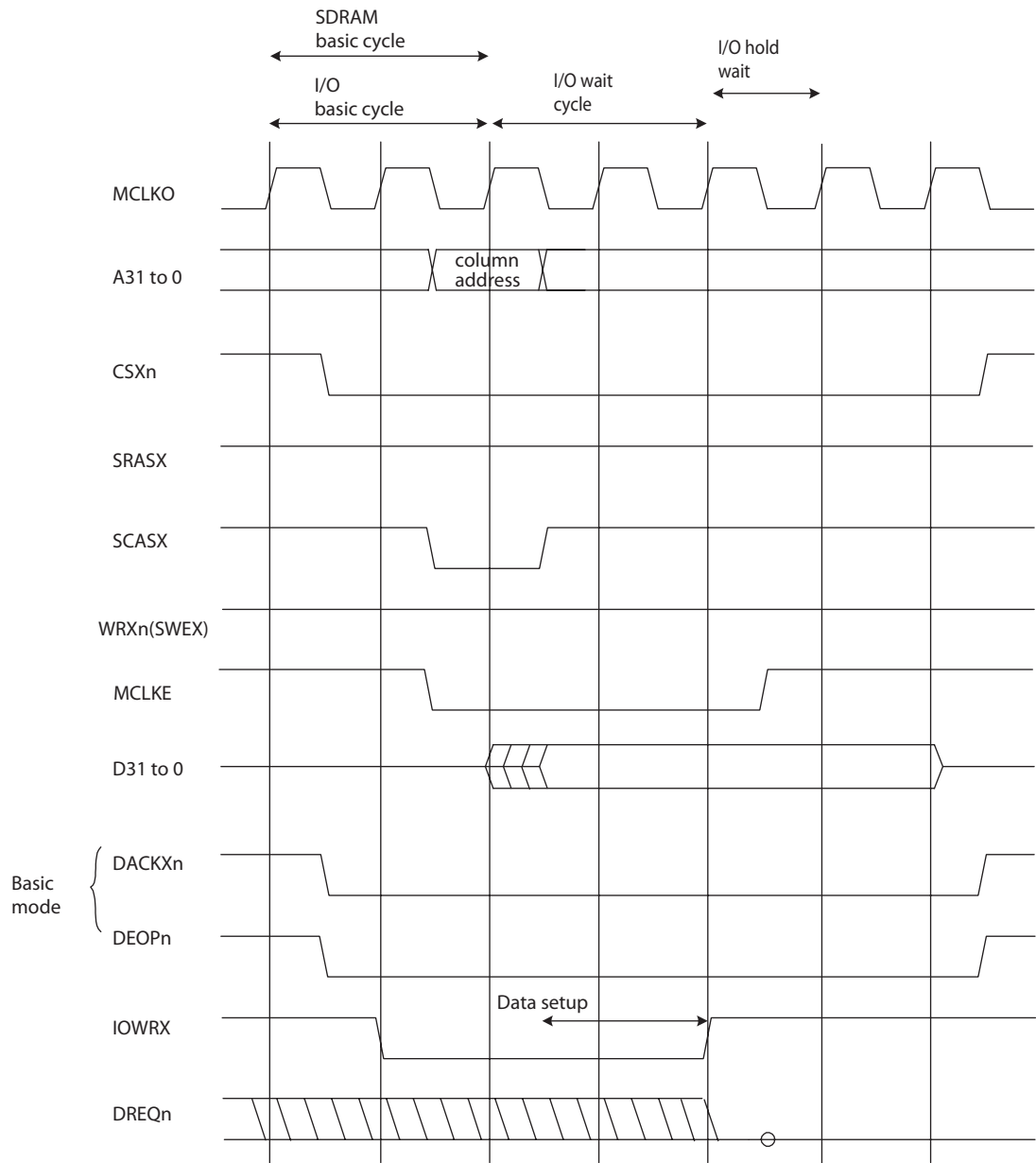
This section describes the operation of DMA fly - by transfer (SDRAM/FCRAM device to I/O).

■ DMA Fly-By Transfer (SDRAM/FCRAM -> I/O)

Figure 31.10-4 shows an operation timing chart assuming TYP3 to TYP0 set to 1000B, AWR set to 0051H, and IOWR set to 42H.

- At SDRAM page hit (Shortest)

Figure 31.10-4 Timing Chart for DMA Fly - by Transfer (SDRAM/FCRAM to I/O) with Page Hits (Shortest)

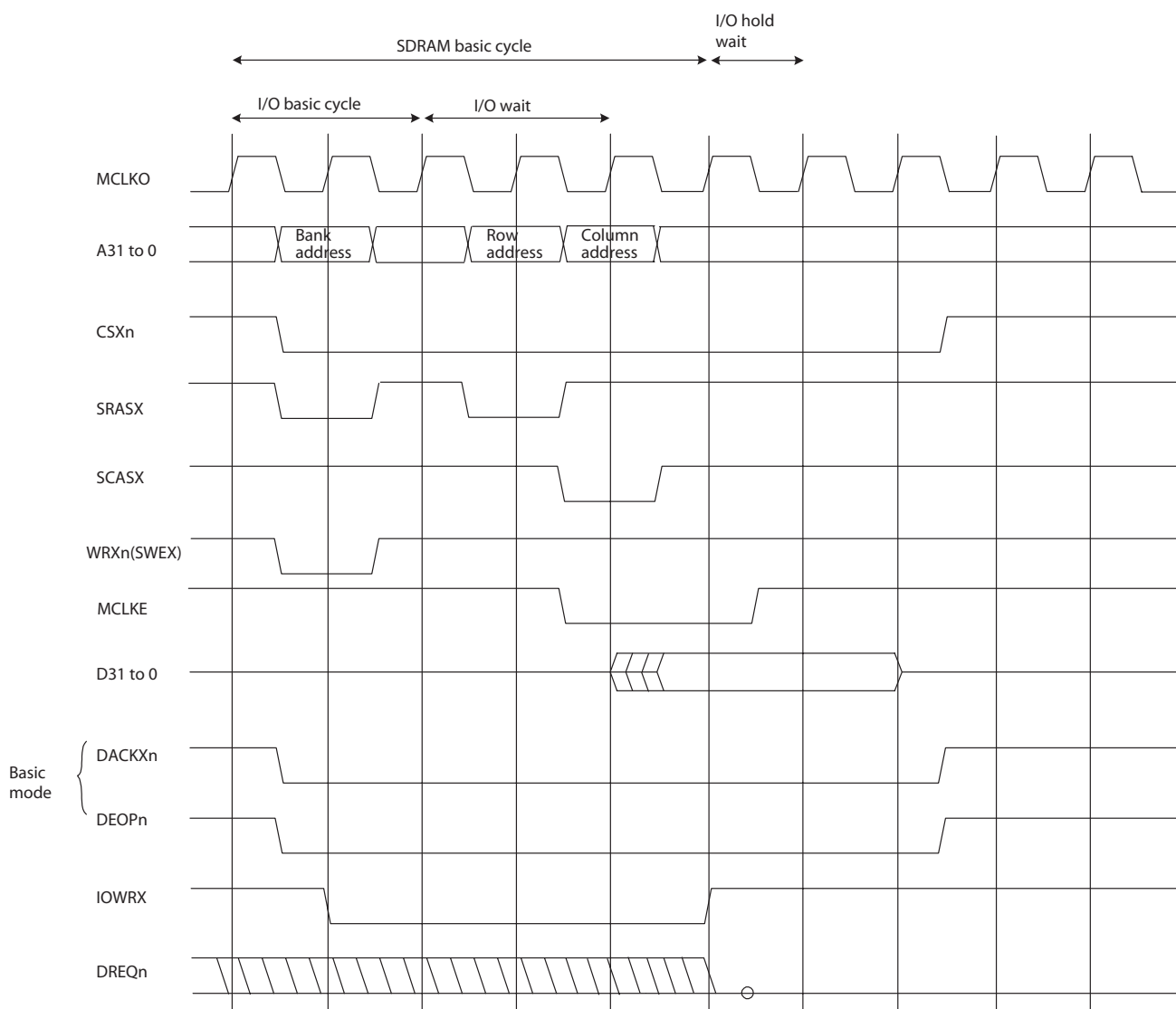


If SDRAM access is shorter than I/O access, the SDRAM access is extended by the I/O access (base access plus I/O wait).

Figure 31.10-5 shows an operation timing chart assuming TYP3 to TYP0 set to 1000B, AWR set to 0051H, and IOWR set to 42H.

● At SDRAM page misses

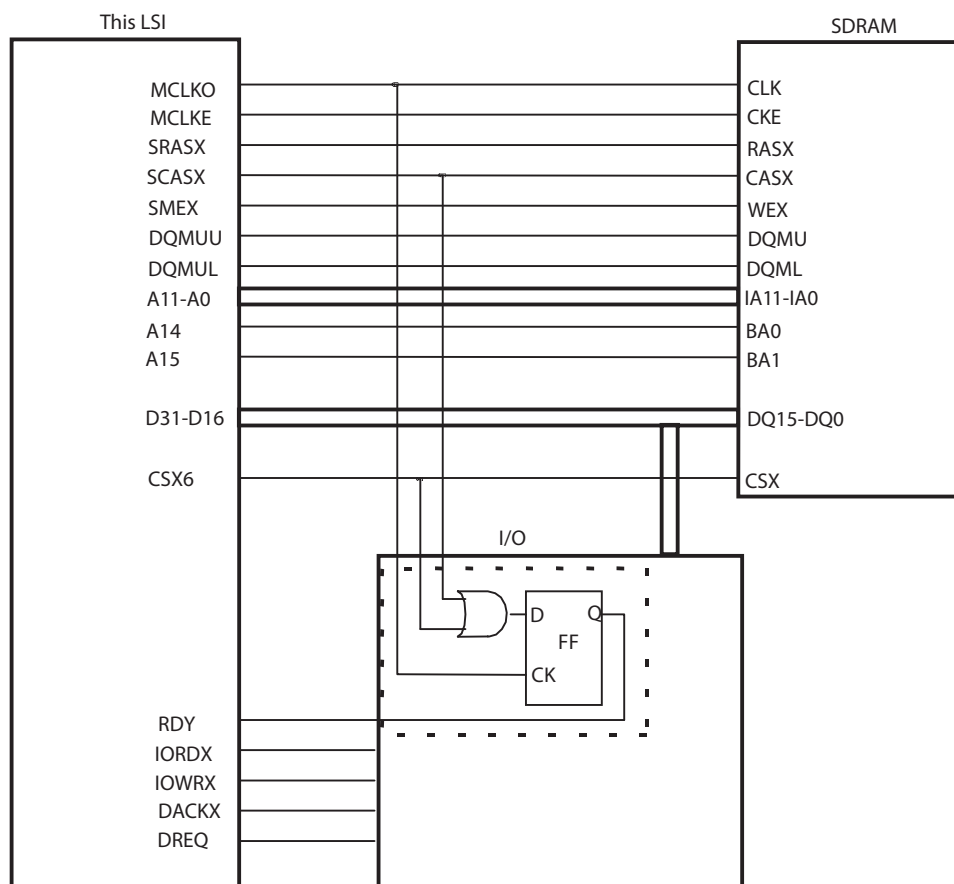
Figure 31.10-5 Timing Chart for DMA Fly - by Transfer (SDRAM/FCRAM to I/O) with Page Misses



- If SDRAM access is extended, for example, by precharging when a page miss occurs in reference to SDRAM, the SDRAM access exceeds the set I/O access, so that the I/O access is extended to be longer than the SDRAM access. When the I/O device requires data setup, therefore, the I/O wait cycle must be set such that I/O access is longer than the maximum SDRAM access cycle. For the above settings, set the number of I/O wait cycles to at least 4.
- For SDRAM/FCRAM on the data output side, a READ command is issued at the timing that satisfies the I/O wait cycle. If the I/O hold cycle has been set, then, a DESL command is issued to insert the I/O hold cycle in the cycle immediately followed by the SDRAM basic cycle.

- For the I/O device on the receiving side, a write strobe of two bus cycles extended by the I/O wait cycle is generated.
- The I/O hold wait cycle does not affect the write strobe.
- Fly - by transfer must always be performed between data buses having the same bus width.
- When the I/O wait cycle is used to reserve data setup time, the I/O wait value must be set according to the page miss condition. A page hit therefore generates a penalty. If this penalty generated at a page hit causes a problem, prepare an external circuit as illustrated in [Figure 31.10-6](#) to use an external wait cycle based on the CAS signal (CL = 2), thereby extending I/O access to reserve data setup time.

Figure 31.10-6 Sample Circuit Solving a Fly-by Penalty Using External Wait Cycles Based on CAS

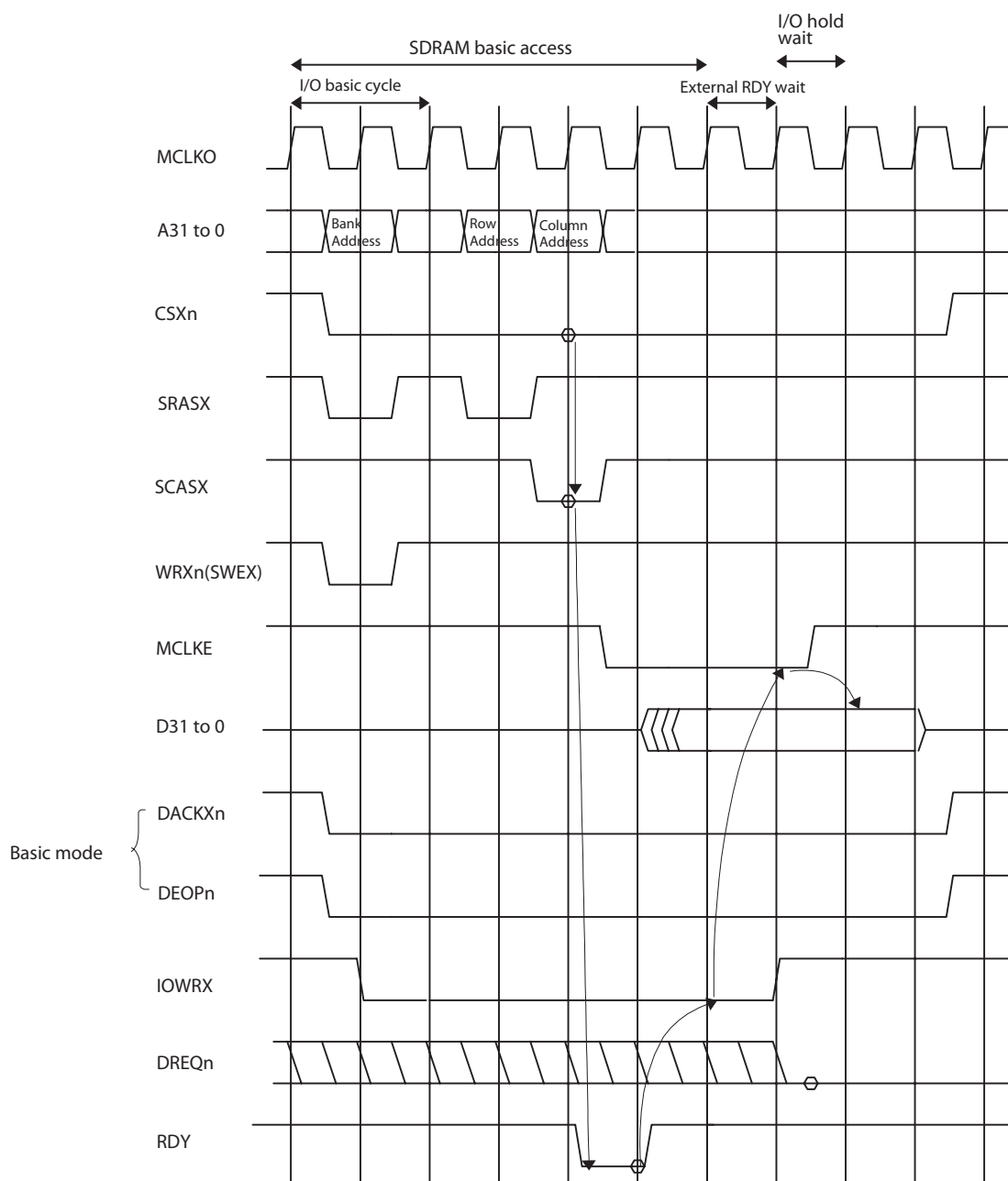


Note:

- For CL = 3, provide two stages of MCLKO - based FF to cause a delay of another cycle.
- If any device requires an external wait cycle, add a logic gate to the RDY signal as required.

Figure 31.10-7 shows the timing chart for the Fly-by Penalty Solution Using External Wait Cycles Based on the CAS Signal (CL = 2):

Figure 31.10-7 Timing Chart for Fly - by Penalty Solution Using External Wait Cycles Based CAS



The rise of the IOWRX signal can be delayed one cycle by extending SDRAM read access one cycle when the signal resulting from OR (negative - logic AND) operation of the CAS signal and the chip select signal for the SDRAM area subject to transfer is input.

As the external wait signal is generated based on the CAS signal rise timing in this case, the data setup time from the SDRAM data output to the I/O device can be reserved for one cycle, regardless of a page hit or miss in SDRAM.

Set the external wait using the RYE0 and RYE1 bits in the DMAC I/O wait register such that the RDY function of

the DMA fly - by access channel to be used is enabled.

When the CAS latency is 3, SDRAM data output is delayed one cycle. Add one stage of FF by the MCLK to input the signal delayed one cycle from the above diagram to the RDY pin.

31.10.5 2-Cycle Transfer (Internal RAM -> External I/O, RAM)

This section explains 2-cycle transfer (internal RAM -> external I/O, RAM) operation.

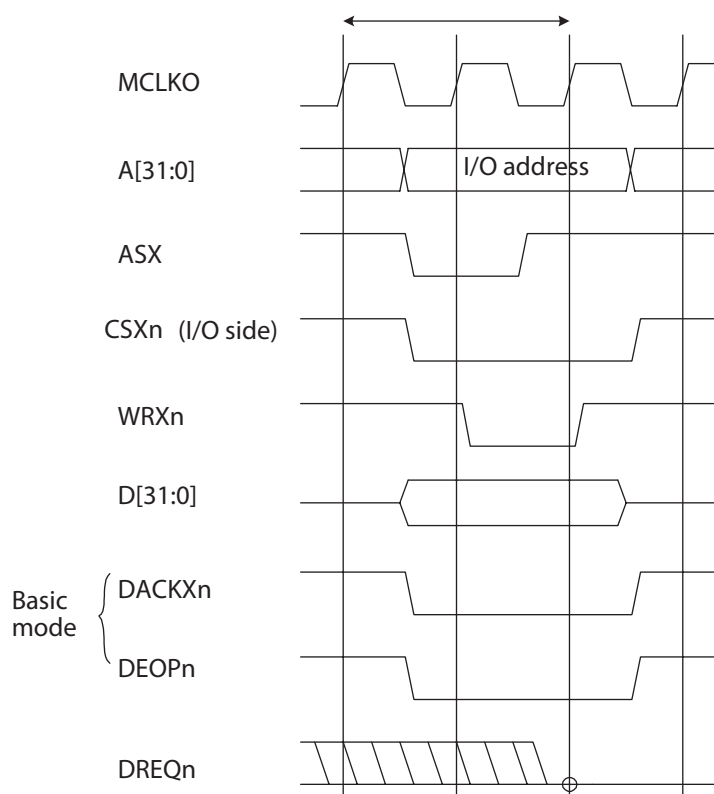
The timing is the same as for external I/O, RAM -> internal RAM.

■ 2-Cycle Transfer (Internal RAM -> External I/O, RAM)

Figure 31.10-8 "Timing Chart for 2-cycle Transfer (Internal RAM -> External I/O, RAM)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=00_H).

Figure 31.10-8 shows a case in which a wait is not set on the I/O side.

Figure 31.10-8 Timing Chart for 2-cycle Transfer (Internal RAM -> External I/O, RAM)



- The bus is accessed in the same way as an interface when DMAC transfer is not performed.
- DACKXn/DEOPn is not output in the internal RAM access cycles.

31.10.6 2-Cycle Transfer (External -> I/O)

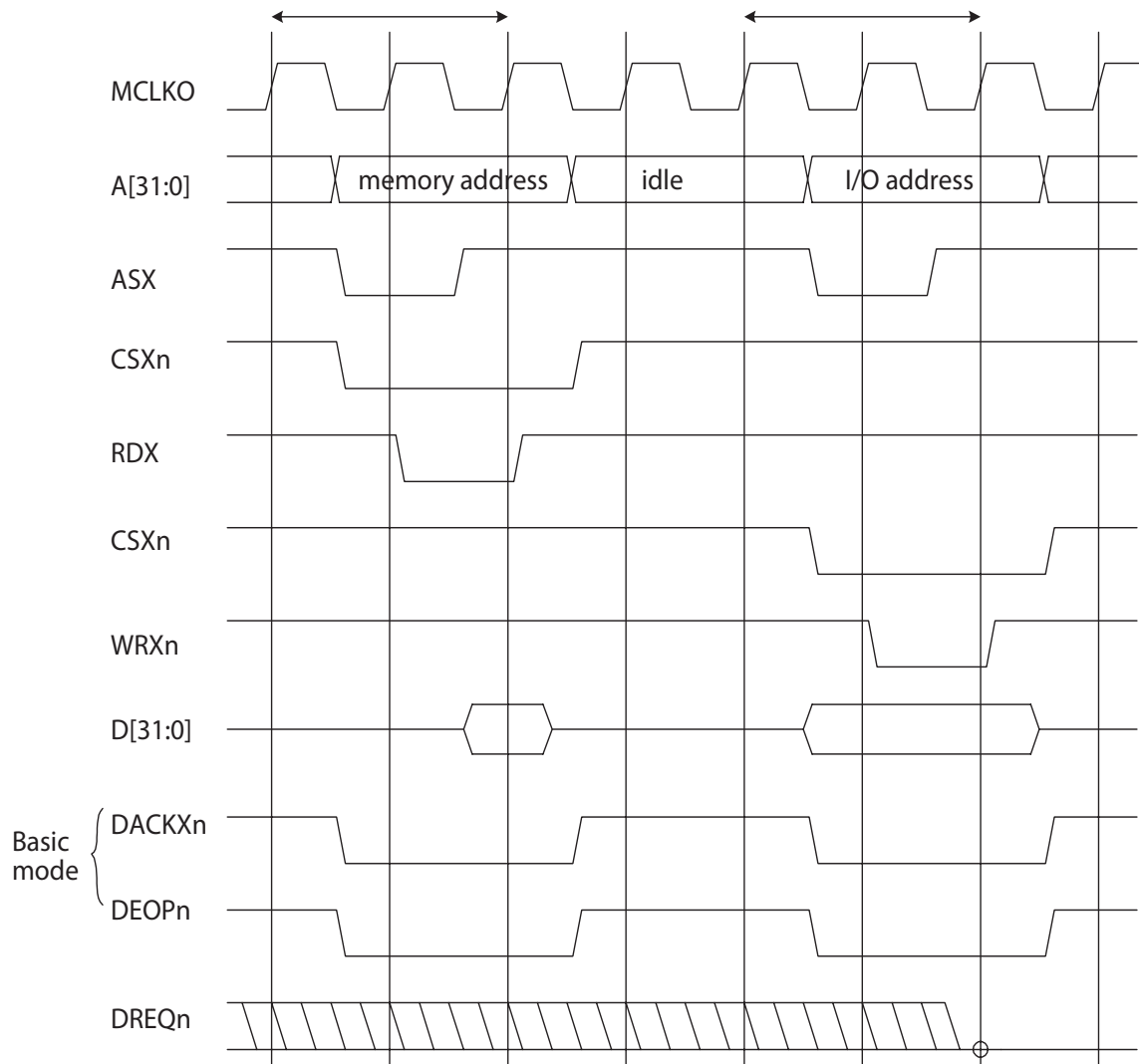
This section explains 2-cycle transfer (external -> I/O) operation.

■ 2-Cycle Transfer (External -> I/O)

Figure 31.10-9 "Timing Chart for 2-Cycle Transfer (External -> I/O)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=00_H).

Figure 31.10-9 shows a case in which a wait is not set for memory and I/O.

Figure 31.10-9 Timing Chart for 2-Cycle Transfer (External -> I/O)



- The bus is accessed in the same way as an interface when the DMAC transfer is not performed.
- In basic mode, DACKXn/DEOPn is output in both transfer source bus access and transfer destination bus access.

31.10.7 2-Cycle Transfer (I/O -> External)

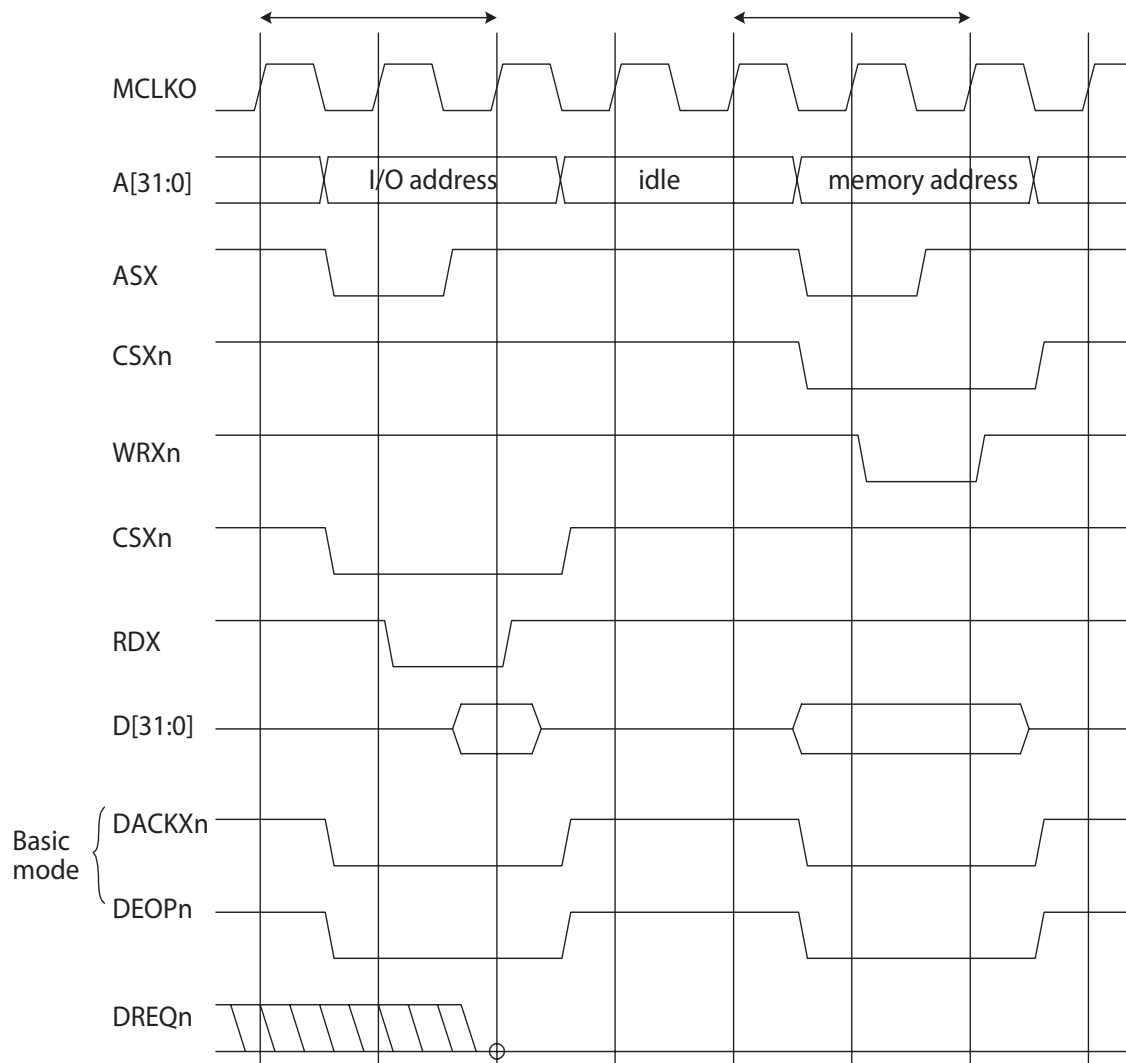
This section explains 2-cycle transfer (I/O -> external) operation.

■ 2-Cycle Transfer (I/O -> External)

Figure 31.10-10 "Timing Chart for 2-Cycle Transfer (I/O -> External)" shows the operation timing chart for (TYP3-0=0000_B, AWR=0008_H, IOWR=00_H).

Figure 31.10-10 shows a case in which a wait is not set for memory and I/O.

Figure 31.10-10 Timing Chart for 2-Cycle Transfer (I/O -> External)



- The bus is accessed in the same way as an interface when the DMAC transfer is not performed.
- In basic mode, DACKXn/DEOPn is output both in the transfer source bus access and transfer destination bus access.

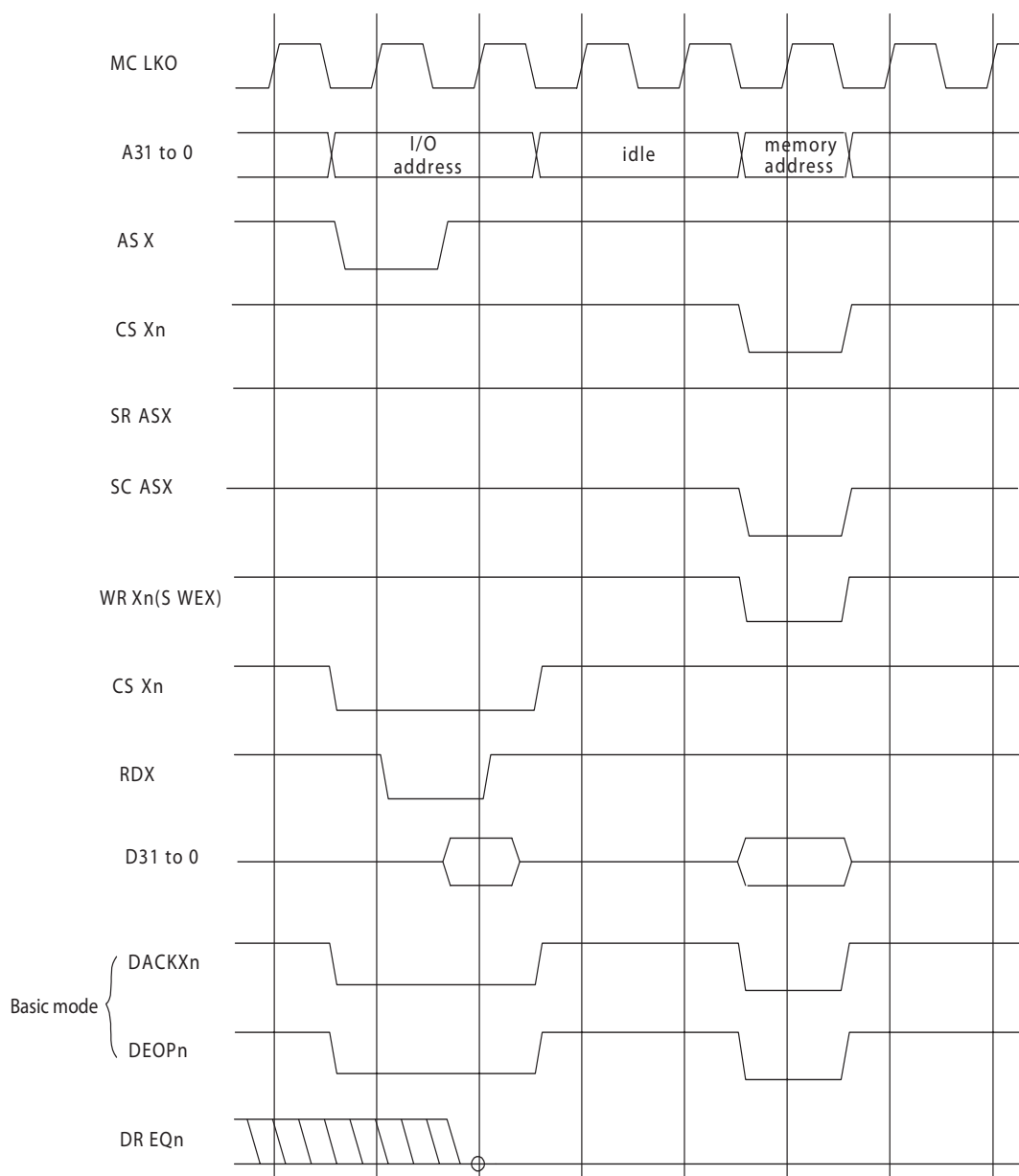
31.10.8 2-Cycle Transfer (I/O -> SDRAM/FCRAM)

This section describes the operation of two - cycle transfer (I/O device to SDRAM/FCRAM).

■ 2-Cycle Transfer (I/O -> SDRAM/FCRAM)

Figure 31.10-11 shows an operation timing chart assuming TYP3 to TYP0 set to 1000B, AWR set to 0051H, and IOWR set to 00H.

Figure 31.10-11 Timing Chart for Two - cycle Transfer (I/O to SDRAM/FCRAM)



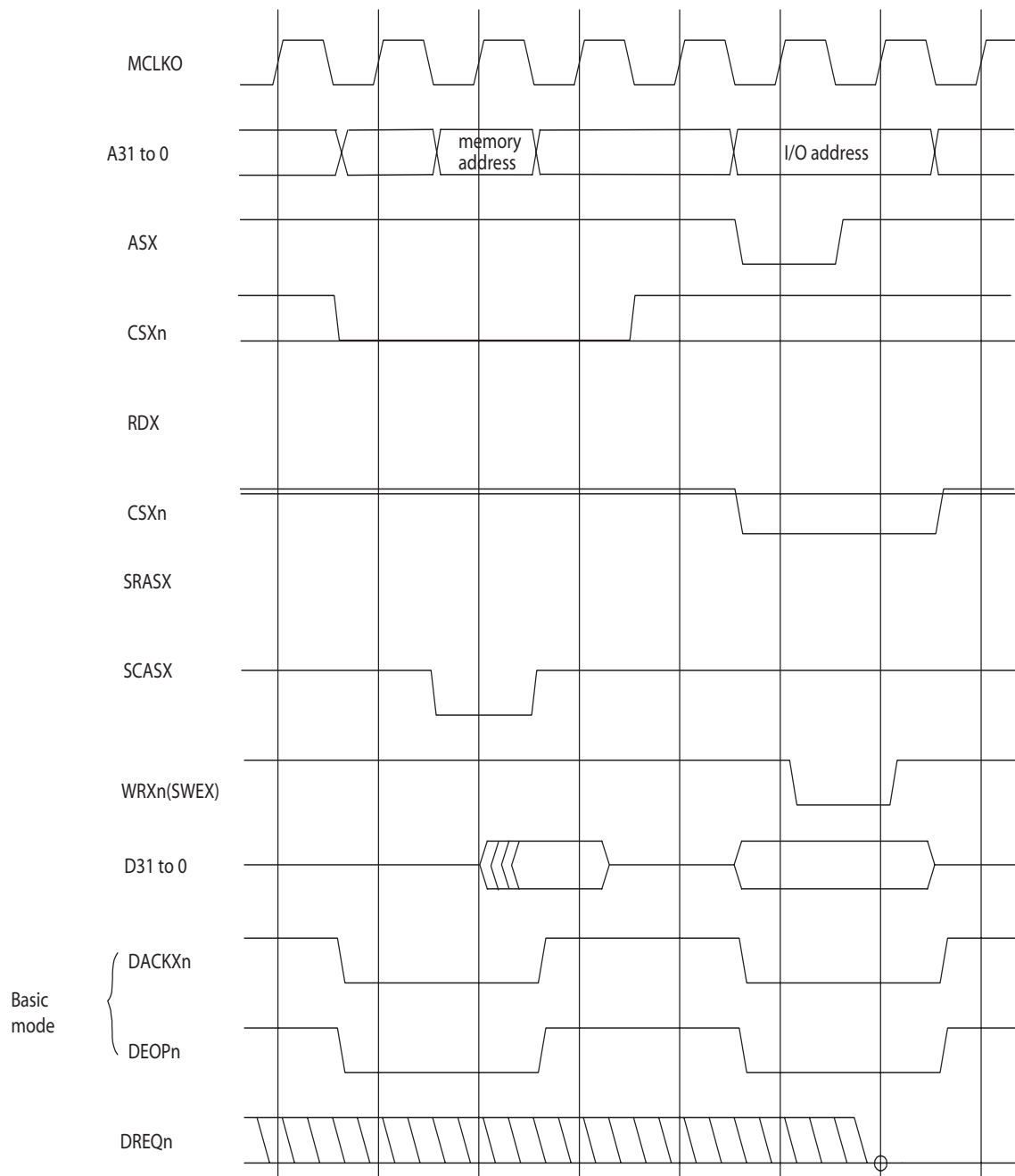
31.10.9 2-Cycle Transfer (SDRAM/FCRAM -> I/O)

This section describes the operation of two - cycle transfer (SDRAM/FCRAM to I/O device).

■ 2-Cycle Transfer (SDRAM/FCRAM -> I/O)

Figure 31.10-12 shows a timing chart for two - cycle transfer (SDRAM/FCRAM to I/O)

Figure 31.10-12 Timing Chart for Two - cycle Transfer (SDRAM/FCRAM to I/O)



- Bus access is the same as that of the interface for non - DMA transfer.
- In base mode, DACKXn/DEOPn is output at both of transfer source bus access and transfer destination bus access.

31.11. Bus Arbitration

This section shows timing charts for releasing the bus right and for acquiring the bus right.

■ Releasing / Acquiring the Bus Right

Figure 31.11-1 "Timing Chart for Releasing the Bus Right" shows the timing chart for releasing the bus right.

Figure 31.11-2 "Timing Chart for Acquiring the Bus Right" shows the timing chart for acquiring the bus right.

Figure 31.11-1 Timing Chart for Releasing the Bus Right

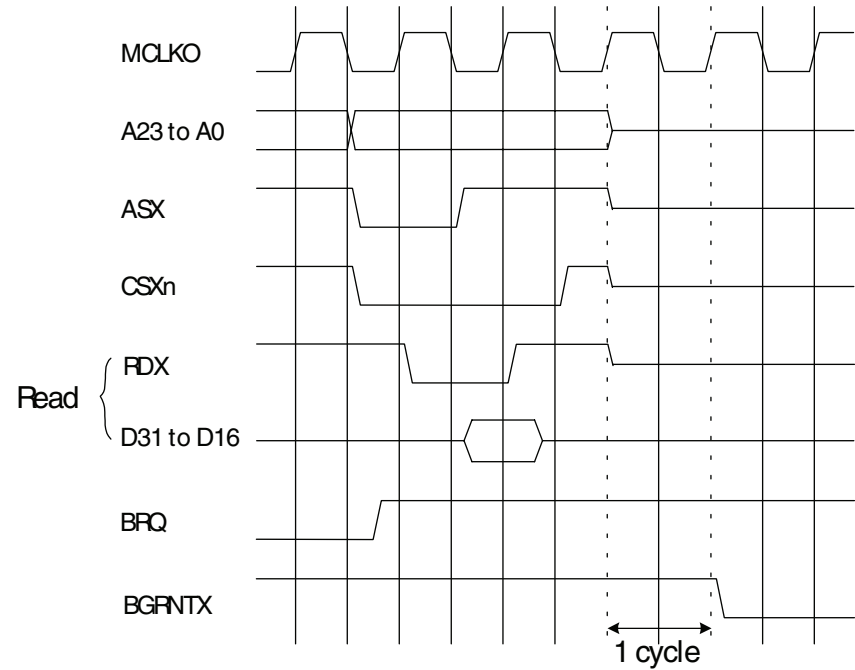
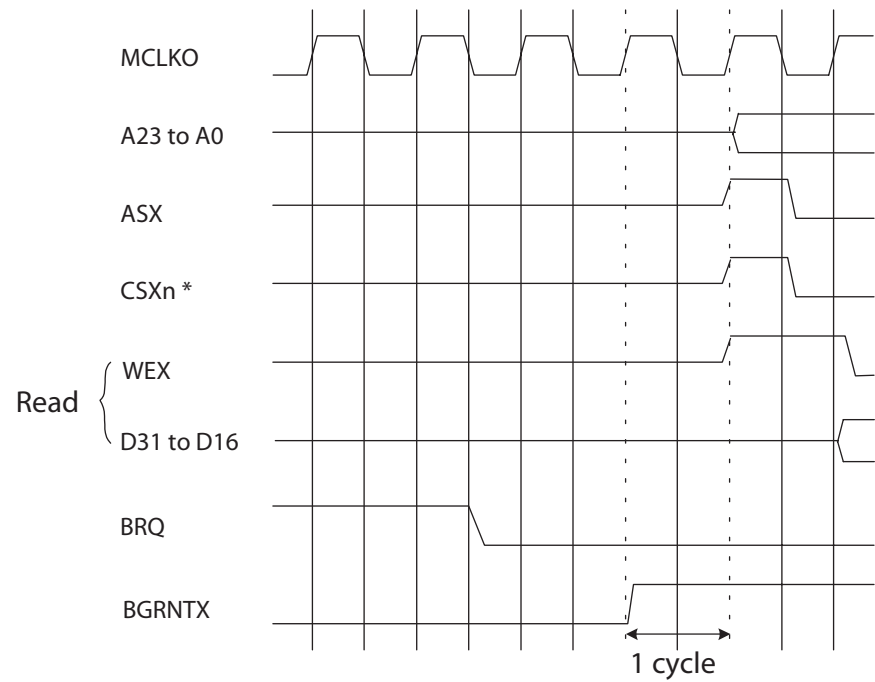


Figure 31.11-2 Timing Chart for Acquiring the Bus Right



- Setting 1 for the BREN bit of the TRC register enables bus arbitration by BRQ/BGRNTX to be performed.
- When the bus right is released, the pin is set to high impedance and then BGRNTX is asserted one cycle later.
- When the bus right is acquired, BGRNTX is negated and then each pin is activated one cycle later.
- CSXn is set to high impedance only if the SREN bit in the ACR0-7 registers is set.
- If all areas enabled by the CSER register are shared (the SREN bit of the ACR register is 1), ASX, BAAX, RDX, WEX, and WRX0-WRX3 are set to high impedance.

31.12. Procedure for Setting a Register

This section explains the procedure for setting a register.

■ Procedure for Setting a Register

Using the following procedures to make external bus interface settings:

1. Before rewriting the contents of a register, be sure to set the CSER register so that the corresponding area is not used (0). If the settings are changed while 1 is set, access before and after the change cannot be guaranteed.
2. Use the following procedure to change a register:
 - Set 0 for the CSER bit corresponding to the applicable area.
 - Set both ASR and ACR at the same time using word access.
 - Set AWR.
 - Set the CHER bit corresponding to the applicable area.
 - Set the CSER bit corresponding to the applicable area.
3. The CSX0 area is enabled after a reset is released. If the area is used as a program area, the register contents need to be rewritten while the CSER bit is 1. In this case, make the settings described in 2) to 4) above in the initial state with a low-speed internal clock. Then, switch the clock to a high-speed clock.
4. Use the following procedure to change the register value in an area for which prefetch:
 - Set 0 for the bit of CSER corresponding to the applicable area.
 - Set 1 for both the PSUS bit and PCLR bit of the TCR register.
 - Set both ASR and ACR at the same time using word access.
 - Set AWR.
 - Set the CHER bit corresponding to the applicable area.
 - Set 0 for both the PSUS bit and PCLR bit of the TCR register.
 - Set 1 for the bit of CSER corresponding to the applicable area.

31.13. Notes on Using the External Bus Interface

This section explains some notes when using the external bus interface.

■ Notes for Use

If settings are made so that the area (TYP3-0=0x0xB) where WRX0-WRX3 are used as a write strobe and the area (TYP3-0=0x1xB) where WEX is used as a write strobe are mixed, be sure to make the following setting in all areas that will be used:

- Set at least one read -> write idle cycle (other than AWR W07-W06=00B).
- Set at least one write recovery cycle (other than AWR W05-W04=00B).

However, if WRX0-WRX3 are disabled (ROM only is connected) in the area (TYP3-0=0x0xB) where WRX0-WRX3 are used as a write strobe, the above restriction does not apply. Also, the above restriction does not apply if both the address -> RDX/WRXn/WEX setup cycle (W01=1) and RDX/WRXn/WEX -> address hold cycle (W00=1) are set in the area (TYP3-0=0x1xB) where WEX is used as a write strobe.

When PFR10_5 is set to '1' the memory clock MCLKO (GP10_4) must be fed back externally to the MCLKI pin (GP10_5). When PFR10_5 is set to '0' the memory clock MCLKO is feed back internally. In this case the MCLKI pin (GP10_5) can be used a general purpose IO. Setting PFR10_5 to '0' (internal feed back) is the recommended operation mode.

MB91460 Series

Chapter 32 USART (LIN / FIFO)

32.1. Overview

This chapter explains the function and operation of the USART. The USART with LIN (Local Interconnect Network) - Function is a general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices. 16 bytes transmission and reception FIFOs are available for selected channels.

The USART provides bidirectional communication function (normal mode), master-slave communication function (multiprocessor mode in master/slave systems), and special features for LIN-bus systems (working both as master or as slave device).

Note: This chapter only lists the registers and addresses of USART04. Please refer to section [3.2. I/O Map \(Page No.36\)](#) for the addresses of the other USARTs.

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

32.1.1 USART Functions

USART is a general-purpose serial data communication interface for transmitting serial data to and receiving data from another CPU or peripheral devices. It has the functions listed in table [32.1-1](#).

Table 32.1-1 USART functions

Item	Function
Data buffer	Full-duplex
Serial Input	5 times oversampling in asynchronous mode
Transfer mode	- Clock synchronous (start-stop synchronization and start-stop-bit-option) - Clock asynchronous (using start-, stop-bits)
Baud rate	- A dedicated baud rate generator is provided, which consists of a 16-bit-reload counter - An external clock can be input and also be adjusted by the reload counter
Data length	- 7 bits (not in synchronous or LIN mode) - 8 bits
Signal mode	Non-return to zero (NRZ) and return to zero (RZ)
Start bit timing	Clock synchronization to the falling edge of the start bit in asynchronous mode
Reception error detection	- Framing error - Overrun error - Parity error
Interrupt request	- Reception interrupt (reception complete, reception error detect, Bus-Idle, LIN-Synch-break detect) - Transmission interrupt (transmission data reg empty) - End of Transmission interrupt ¹

Table 32.1-1 USART functions (continued)

Item	Function
Master-slave communication function (multiprocessor mode)	One-to-n communication (one master to n slaves) (This function is supported both for master and slave system).
Synchronous mode	Function as Master- or Slave-USART
Transceiving pins	Direct access possible
LIN bus options	<ul style="list-style-type: none"> - Operation as master device - Operation as slave device - Generation of LIN-Sync-break - Detection of LIN-Sync-break - Detection of start/stop edges in LIN-Sync-field connected to ICU 0 and 2
Synchronous serial clock	The synchronous serial clock can be output continuously on the SCK pin for synchronous communication with start & stop bits
Clock delay option	Special synchronous Clock Mode for delaying clock (useful for SPI-compliance)
16 byte deep FIFO	FIFO can be activated with receive programmable trigger level

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

32.1.2 USART operation modes

The USART operates in four different modes, which are determined by the MD0- and the MD1-bit of the Serial mode register (SMR04). Mode 0 and 2 are used for bidirectional serial communication, mode 1 for master/slave communication and mode 3 for LIN master/slave communication.

Table 32.1-2 USART operation modes

Operation mode		Data length		Synchroniza- tion of mode	Length of stop bit	data bit direction*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first.

** "+1" means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

Mode 1 operation is supported both for master or slave operation of USART in a master-slave connection system. In Mode 3 the USART function is locked to 8N1-Format, LSB first.

If the mode is changed, USART cuts off all possible transmission or reception and then awaits new action. The MD1 and MD0 bits of the Serial Mode Register (SMR04) determine the operation mode of USART04 as shown in the following table:

Table 32.1-3 Mode Bit Setting

MD1	MD0	Mode	Description
0	0	0	Asynchronous (normal mode)
0	1	1	Asynchronous (multiprocessor mode)
1	0	2	Synchronous (normal mode)
1	1	3	Asynchronous (LIN mode)

32.1.3 USART Interrupts

Table 32.1-4 shows the interrupts of USART4.

Table 32.1-4 USART interrupts

Interrupt cause	Interrupt number	Interrupt control register		Interrupt Vector	
		Register name	Address	Offset	Default address
USART04 reception (RX)	#66 (42 _H)	ICR25	0459 _H	2F4 _H	000FFE4 _H
USART04 transmission (TX) USART04 end of transmission (EoT) ¹	#67 (43 _H)	ICR25	0459 _H	2F0 _H	000FFE0 _H

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

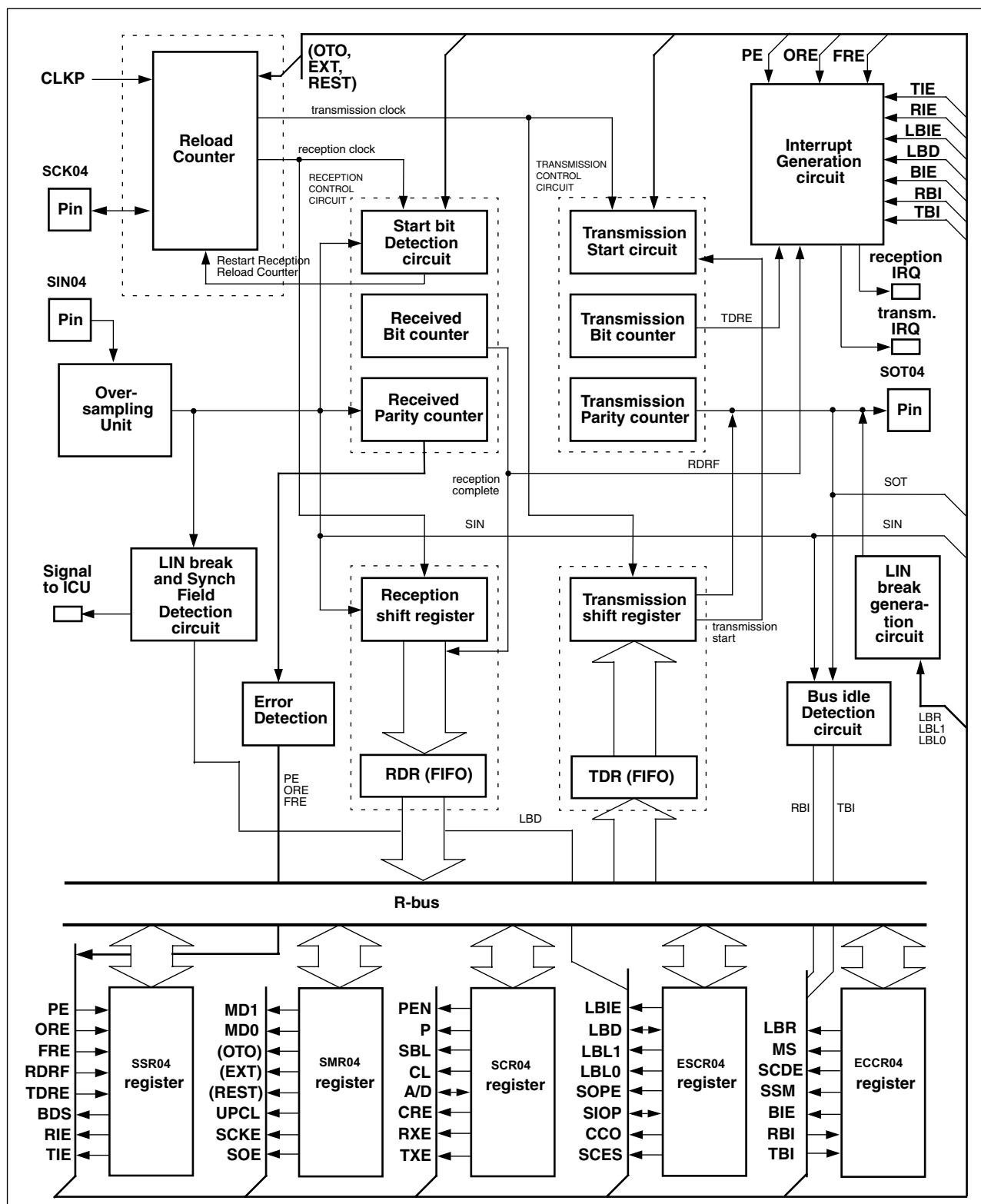
Please refer to section 3.3. [Interrupt Vector Table \(Page No.113\)](#) for the interrupts of the other USARTs.

32.2. USART Configuration

■ USART consists of the following blocks:

- Reload Counter
- Reception Control Circuit
- Reception Shift Register
- Reception Data Register (RDR)
- Transmission Control Circuit
- Transmission Shift Register
- Transmission Data Register (TDR)
- Error Detection Circuit
- Oversampling Unit
- Interrupt Generation Circuit
- LIN Break Generation
- LIN Break and Synch Field Detection
- Bus Idle Detection Circuit
- Serial Mode Register (SMR)
- Serial Control Register (SCR)
- Serial Status Register (SSR)
- Extended Com. Contr. Reg. (ECCR)
- Extended Status/Contr. Reg. (ESCR)
- FIFO Control Register (FCR)
- FIFO Status Register (FSR)

Figure 32.2-1 USART Block Diagram



■ Explanation of the different blocks

- Reload Counter

The reload counter functions as the dedicated baud rate generator. It can select external input clock or CLKP for the transmitting and receiving clocks. The reload counter has a 16 bit register for the reload value. The actual count of the transmission reload counter can be read via the BGR0/1 registers.

- Reception Control Circuit

The reception control circuit consists of a received bit counter, start bit detection circuit, and received parity counter. The received bit counter counts reception data bits. When reception of one data item for the specified data length is complete, the received bit counter sets the Reception data register full flag. When the FIFO is enabled, the flag is set if the triggerlevel is reached. The start bit detection circuit detects start bits from the serial input signal and sends a signal to the reload counter to synchronize it to the falling edge of these start bits. The reception parity counter calculates the parity of the reception data.

- Reception Shift Register

The reception shift register fetches reception data input from the SIN04 pin, shifting the data bit by bit. When reception is complete, the reception shift register transfers receive data to the RDR04 register.

- Reception Data Register

This register retains reception data. Serial input data is converted and stored in this register. If the FIFO is enabled up to 16 receptions can be saved, the trigger level is programmable.

- Transmission Control Circuit

The transmission control circuit consists of a transmission bit counter, transmission start circuit, and transmission parity counter. The transmission bit counter counts transmission data bits. When the transmission of one data item of the specified data length is complete, the transmission bit counter sets the Transmission data register full flag. The transmission start circuit starts transmission when data is written to TDR04. The transmission parity counter generates a parity bit for data to be transmitted if parity is enabled.

- Transmission Shift Register

The transmission shift register transfers data written to the TDR04 register to itself and outputs the data to the SOT04 pin, shifting the data bit by bit.

- Transmission Data Register

This register sets transmission data. Data written to this register is converted to serial data and output.

If the FIFO is enabled up to 16 transmissions can be saved and continuously transmitted

- Error Detection Circuit

The error detection circuit checks if there was any error during the last reception. If an error has occurred it sets the corresponding error flags.

- Oversampling Unit

The oversampling unit oversamples the incoming data at the SIN04 pin for five times. It is switched off in synchronous operation mode.

- Interrupt Generation Circuit

The interrupt generation circuit administers all cases of generating a reception or transmission interrupt. If a corresponding request enable bit is set and an interrupt case occurs the interrupt request will be generated immediately.

- LIN Break and Synchronization Field Detection Circuit

The LIN break and LIN synchronization field detection circuit detects a LIN break, if a LIN master node is sending a message header. If a LIN break is detected a special flag bit is generated. The first and the fifth falling edge of the synchronization field is recognized by this circuit by generating an internal signal for the Input Capture Unit to measure the actual serial clock time of the transmitting master node.

- LIN Break Generation Circuit

The LIN break generation circuit generates a LIN break of a determined length.

- Bus Idle Detection circuit

The bus idle detection circuit recognizes if neither reception nor transmission is going on. In this case the circuit generates a special flag bit.

- Serial Mode Register

This register performs the following operations:

- Selecting the USART operation mode
- Selecting a clock input source
- Selecting if an external clock is connected “one-to-one” or connected to the reload counter
- Resetting the USART (preserving the settings of the registers)
- Specifying whether to enable serial data output to the corresponding pin
- Specifying whether to enable clock output to the corresponding pin

- Serial Control Register

This register performs the following operations:

- Specifying whether to provide parity bits
- Selecting parity bits
- Specifying a stop bit length
- Specifying a data length
- Selecting a frame data format in mode 1
- Clearing the error flags
- Specifying whether to enable transmission
- Specifying whether to enable reception

- Serial Status Register

This register indicates the transmission and reception status and error status, and enables/disables transmission and reception interrupt requests.

- Extended Status/Control Register

This register provides several LIN functions, direct access to the SIN04 and SOT04 pin and setting for the USART synchronous clock mode.

- Extended Communication Control Register

The extended communication control register provides bus idle recognition interrupt settings, synchronous clock settings, and the LIN break generation.

- FIFO Control Register

With the FCR4 register the TX/RX FIFOs can be enabled, the RX interrupt triggerlevel can be set and the FIFO status register mode can be set.

- FIFO Status Register

With the FSR4 register shows the number of valid RX/TX data in the FIFO buffers.

Additionally, the FSR register handles the End of Transmission (EoT) interrupt ¹.

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

32.3. USART Pins

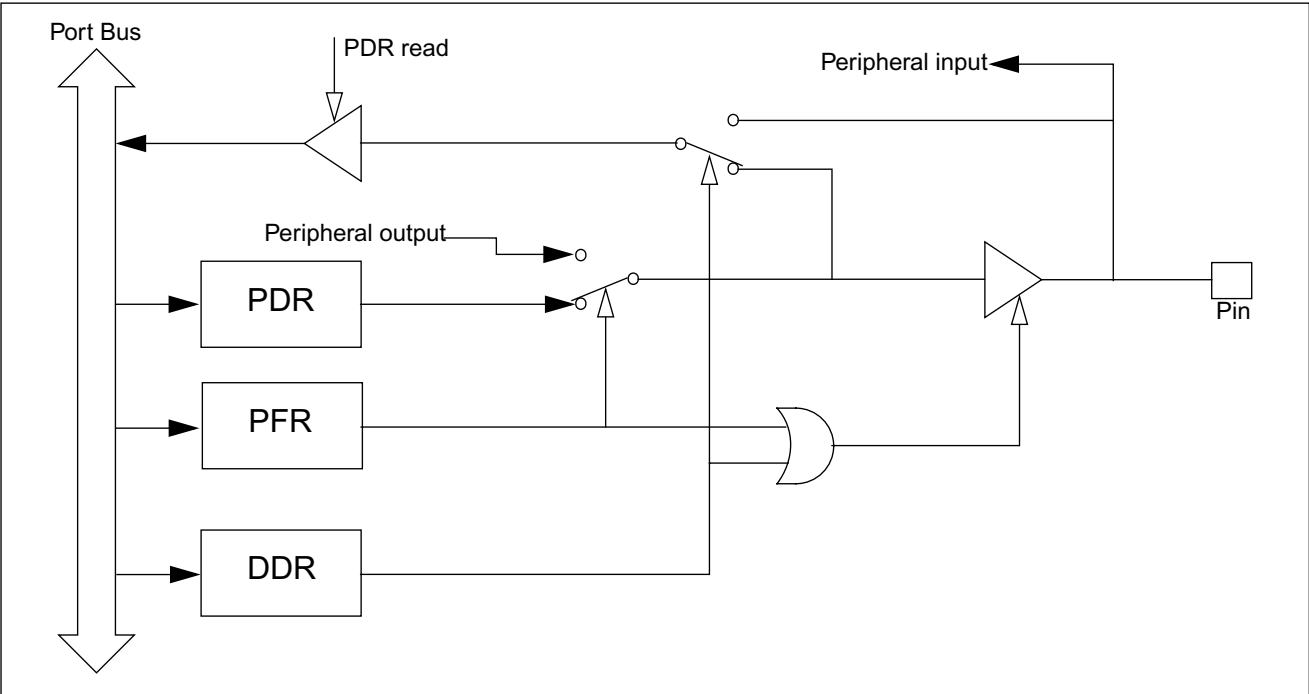
■ USART Pins

The USART pins also serve as general ports. Table 32.3-1 lists the pin functions, I/O formats, and settings required to use USART.

Table 32.3-1 USART04 Pins

Pin name	Pin function	I/O format	Pull-up Pull-down	Standby control	Setting required to use pin
SIN04	Port I/O or serial data input	CMOS output and CMOS hystere- sis, CMOS Auto- motive hysteresis, TTL input	Program- mable	Provided	Set port function mode: PFR: bit0 = 1, EPFR: bit0 = 0
SOT04	Port I/O or serial data output				Set to output enable mode: SMR04: SOE = 1, Set port function mode: PFR: bit1 = 1 EPFR: bit1 = 0
SCK04	Port I/O or serial clock input/output				Set port function mode: PFR: bit2 = 1 EPFR: bit2 = 0 Set to output enable mode when a clock is output SMR04: SCKE = 1

Figure 32.3-1 Block Diagram of USART pins



MB91460 Series**32.4. USART Registers**

The following table defines the USART04 registers:

Table 32.4-1 USART04 Registers

Address	bit 15	bit 8	bit 7	bit 0
060 _H , 061 _H	SCR04 (Serial Control Register)		SMR04 (Serial Mode Register)	
062 _H , 063 _H	SSR04 (Serial Status Register)		RDR04/TDR04 (Rx, Tx Data Register)	
064 _H , 065 _H	ESCR04 (Extended Status/Control Reg.)		ECCR04 (Extended Comm. Contr. Reg.)	
066 _H , 067 _H	FSR04 (FIFO status register)		FCR04 (FIFO control register)	
088 _H , 089 _H	BGR104 (Baud Rate Generator Reg. 1)		BGR004 (Baud Rate Generator Reg. 0)	

Note: FSR (FIFO status register) and FCR (FIFO control register) register are only available on USARTs with FIFO option, i.e. USART ch. 4-7.

On devices with End of Transmission interrupt ¹, FSR is always available.

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

32.4.1 Serial Control Register 04 (SCR04)

This register specifies parity bits, selects the stop bit and data lengths, selects a frame data format in mode 1, clears the reception error flag, and specifies whether to enable transmission and reception.

Figure 32.4-1 Serial Control Register 04 (SCR04)

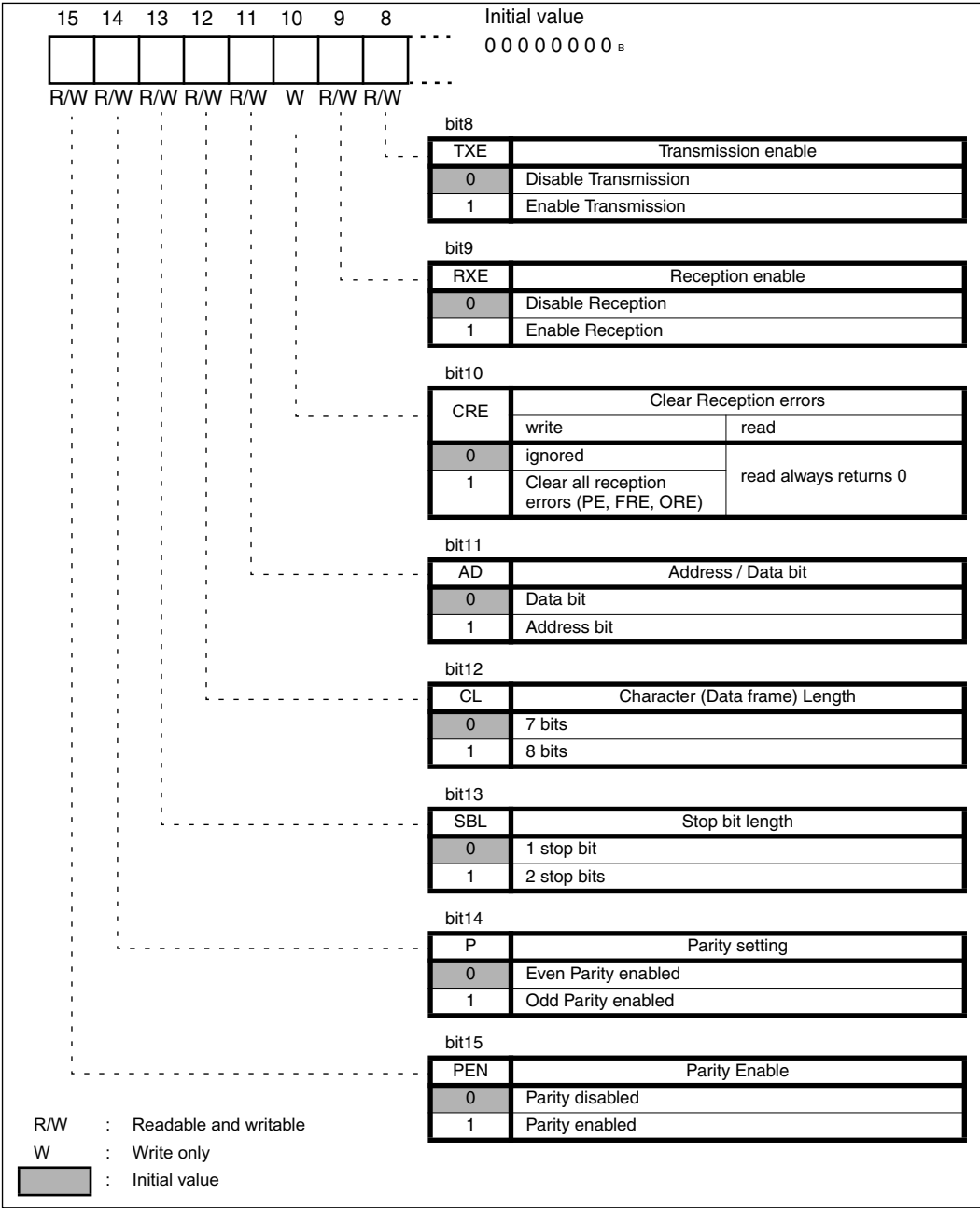


Table 32.4-2 Functions of each bit of control register 04 (SCR04)

Bit name		Function
bit15	PEN: Parity enable bit	This bit selects whether to add a parity bit during transmission in serial asynchronous mode or detect it during reception. Parity is only provided in mode 0 and in mode 2 if SSM of the ECCR04 is selected. This bit is fixed to 0 (no parity) in mode 3 (LIN).
bit14	P: Parity selection bit	When parity is provided and enabled this bit selects even (0) or odd (1) parity
bit13	SBL: Stop bit length selection bit	This bit selects the length of the stop bit of an asynchronous data frame or a synchronous frame if SSM of the ECCR04 is selected. This bit is fixed to 0 (1 stop bit) in mode 3 (LIN).
bit12	CL: Data length selection bit	This bit specifies the length of transmission or reception data. This bit is fixed to 1 (8 bits) in mode 2 and 3.
bit11	AD: Address/Data selection bit *	This bit specifies the data format in multiprocessor mode 1. Writing to this bit determines an address or data frame to be sent next, reading from it returns the last received kind of frame. "1" indicates an address frame, "0" indicates a usual data frame. Note: During a RMW-Read cycle the AD bit returns the value to be sent instead of the last received AD bit. see table below*
bit10	CRE: Clear reception error flags bit	This bit clears the FRE, ORE, and PE flag of the Serial Status Register (SSR04). This bit also clears a possible reception interrupt caused by errors. Writing a 1 to it clears the error flag. Writing a 0 has no effect. Reading from it always returns 0.
bit9	RXE: Reception enable bit	This bit enables USART reception. If this bit is set to 0, USART disables the reception of data frames. The LIN break detection in mode 0 or 3 remains unaffected.
bit8	TXE: Transmission enable bit	This bit enables USART transmission. If this bit is set to 0, USART disables the transmission of data frames.

* see table 32.4-3 for R/W options

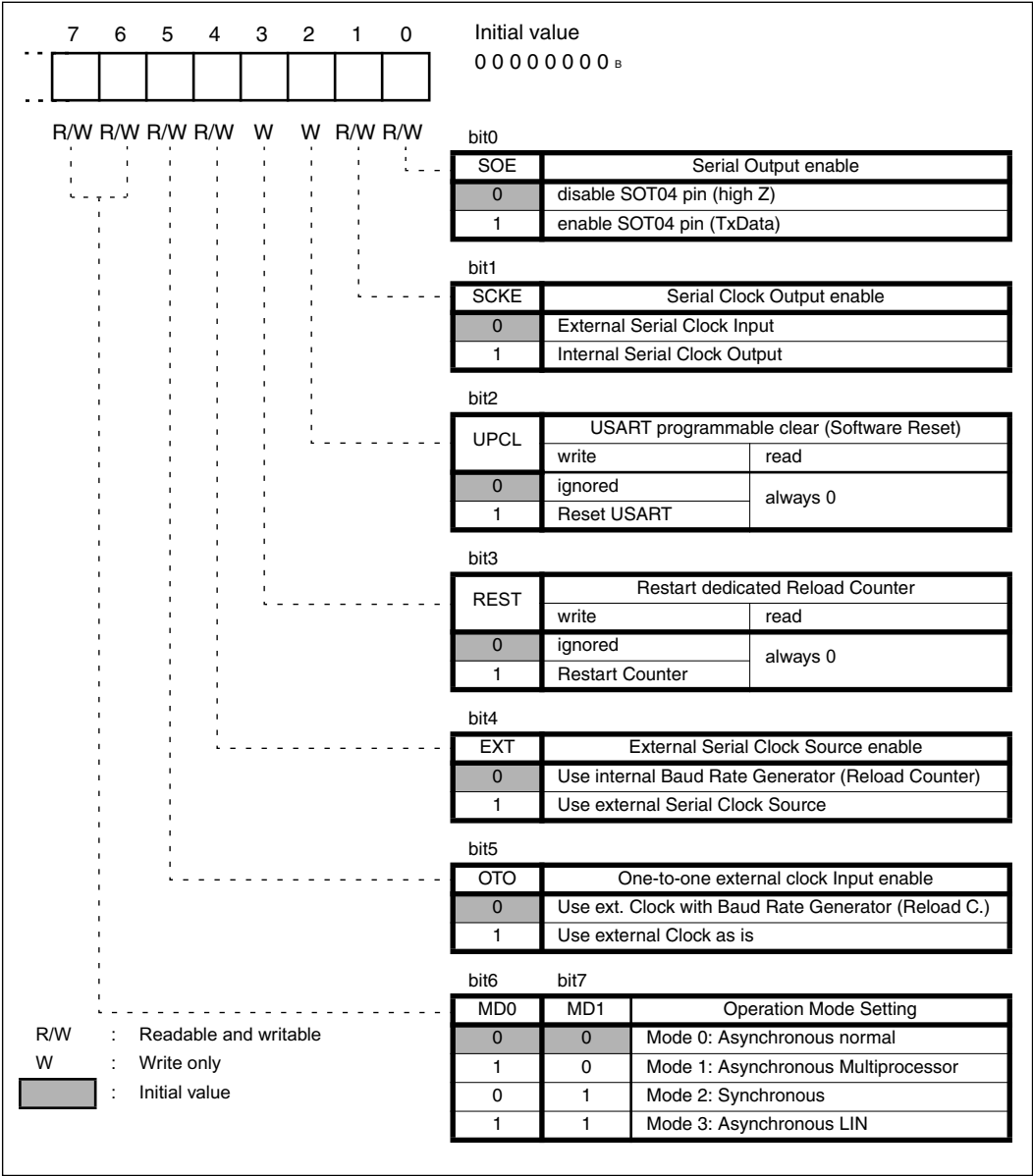
Table 32.4-3 * Read/Write options of AD-Bit

Cycle	Action
Write	Write data to be sent to AD-Bit
Normal Read	Read received AD-Bit
RMW-Read	Read data to be sent from AD-Bit

32.4.2 Serial Mode Register 04 (SMR04)

This register selects an operation mode and baud rate clock and specifies whether to enable output of serial data and clocks to the corresponding pin.

Figure 32.4-2 Configuration of the Serial Mode register 04 (SMR04)



MB91460 Series**Table 32.4-4 Bit function of the Serial Mode register 04 (SMR04)**

Bit name		Function
bit7 bit6	MD1 and MD0: Operation mode selection bits	These two bits sets the USART operation mode.
bit5	OTO: One-to-one external clock selection bit	This bit sets an external clock directly to the USART's serial clock. This function is used for synchronous slave mode operation
bit4	EXT: External clock selection bit	This bit executes internal or external clock source for the reload counter. (Internal clock source: CLKP)
bit3	REST: Restart of transmission reload counter bit	If a 1 is written to this bit the reload counter is restarted. Writing 0 to it has no effect. Reading from this bit always returns 0.
bit2	UPCL: USART programmable clear bit (Software reset)	Writing a 1 to this bit resets USART immediately. The register settings are preserved. Possible reception or transmission will cut off. All error flags are cleared and the Reception Data Register (RDR04) contains 00h. Writing 0 to this bit has no effect. Reading from it always returns 0.
bit1	SCKE: Serial clock output enable	<ul style="list-style-type: none"> • This bit controls the serial clock input-output ports. • When this bit is 0, the SCK04 pin operates as serial clock input pin. When this bit is 1, the SCK04 pin operates as serial clock output pin. <Caution> <ul style="list-style-type: none"> • When using the SCK04 pin as serial clock input (SCKE=0) pin, set the pin as input port. Also, select external clock (EXT = 1) using the external clock selection bit.
bit0	SOE: Serial data output enable bit	<ul style="list-style-type: none"> • This bit enables or disables the output of serial data. • When this bit is 0, the SOT04 pin outputs the default mark level. When this bit is 1, the SOT04 pin outputs the transmission data.

32.4.3 Serial Status Register 04 (SSR04)

This register checks the transmission status, reception status and error status, and enables and disables the transmission and reception interrupt requests.

Figure 32.4-3 Configuration of the Serial Status register 04 (SSR04)

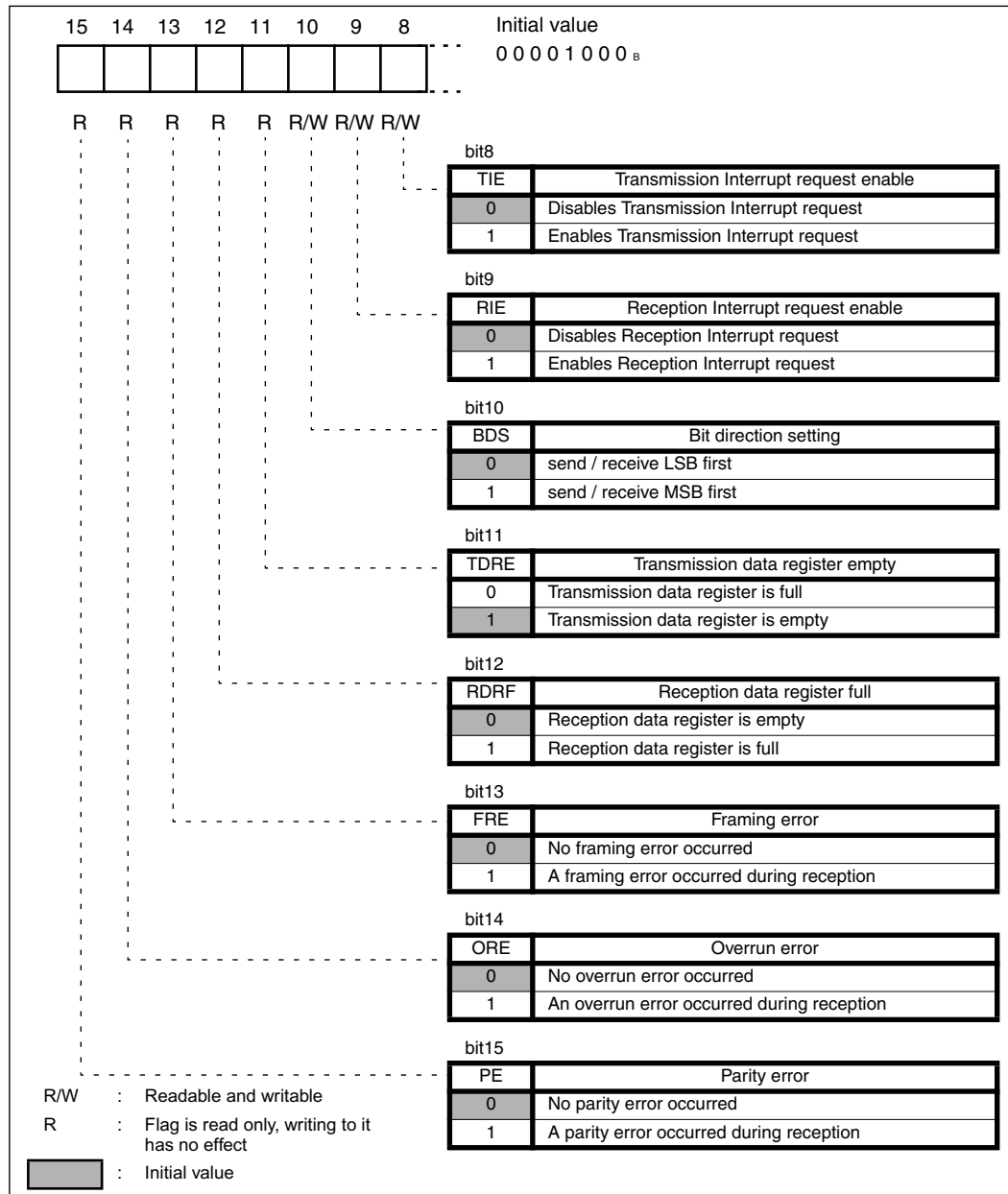


Table 32.4-5 Functions of each bit of status register 04 (SSR04)

Bit name		Function
bit15	PE: Parity error flag	<ul style="list-style-type: none"> This flag is set to 1 when a parity error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register (SCR04). A reception interrupt request is output when this flag and the RIE bit are 1. Data in the reception data register (RDR04) is invalid when this flag is set.
bit14	ORE: Overrun error flag	<ul style="list-style-type: none"> This flag is set to 1 when an overrun error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register (SCR04). A reception interrupt request is output when this flag and the RIE bit are 1. Data in the reception data register (RDR04) is invalid when this flag is set.
bit13	FRE: Framing error flag	<ul style="list-style-type: none"> This flag is set to 1 when a framing error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register 1 (SCR04). A reception interrupt request is output when this flag and the RIE bit are 1. Data in the reception data register (RDR04) is invalid when this flag is set.
bit12	RDRF: Receive data full flag	<ul style="list-style-type: none"> This flag indicates the status of the reception data register (RDR04). This flag is set to 1 when reception data is loaded into RDR04 and can only be cleared to 0 when the reception data register (RDR04) is read. A reception interrupt request is output when this flag and the RIE bit are 1.
bit11	TDRE: Transmission data empty flag	<ul style="list-style-type: none"> This flag indicates the status of the transmission data register (TDR04). This flag is cleared to 0 when transmission data is written to TDR04 and is set to 1 when data is loaded into the transmission shift register and transmission starts. A transmission interrupt request is generated if this flag and the TIE bit are 1. <p><Caution> This flag is set to 1 (TDR04 empty) as its initial value.</p>
bit10	BDS: Transfer direction selection bit	<ul style="list-style-type: none"> This bit selects whether to transfer serial data from the least significant bit (LSB first, BDS=0) or the most significant bit (MSB first, BDS=1). <p><Caution> The high-order and low-order sides of serial data are interchanged with each other during reading from or writing to the serial data register. If this bit is set to another value after the data is written to the RDR04 register, the data becomes invalid.</p> <p>This bit is fixed to 0 in mode 3 (LIN)</p>
bit9	RIE: Reception interrupt request enable bit	<ul style="list-style-type: none"> This bit enables or disables input of a interrupt request for transmission to the CPU. A reception interrupt request is output when this bit and the reception data flag (RDRF) are 1 or this bit and one or more error flags (PE, ORE, and FRE) are 1.

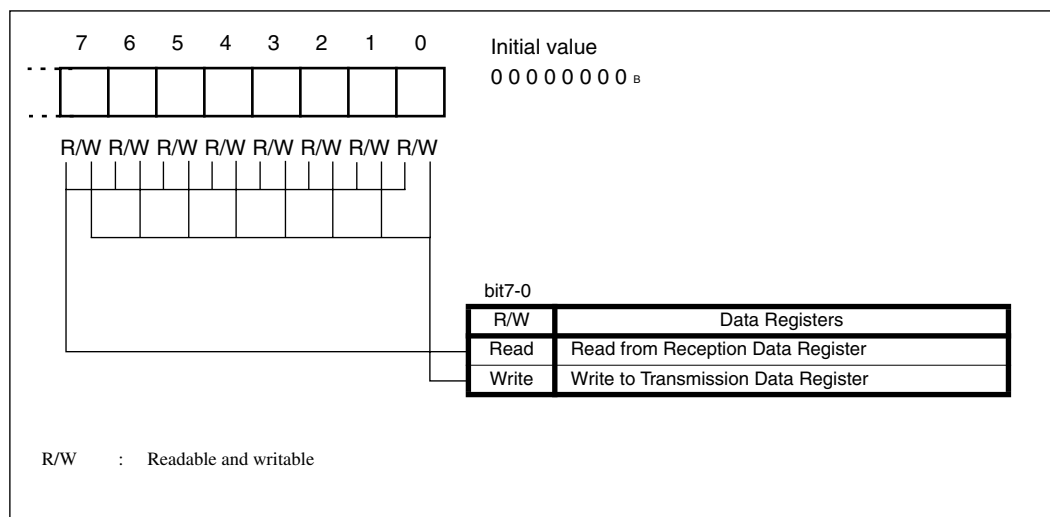
Table 32.4-5 Functions of each bit of status register 04 (SSR04) (continued)

Bit name		Function
bit8	TIE: Transmission interrupt request enable bit	<ul style="list-style-type: none"> This bit enables or disables output of a request for transmission interrupt to the CPU. A transmission interrupt request is output when this bit and the TDRE bit are 1.

32.4.4 Reception and Transmission Data Register (RDR04 / TDR04)

The reception data register (RDR04) holds the received data. The transmission data register (TDR04) holds the transmission data. Both RDR04 and TDR04 registers are located at the same address.

Note: TDR04 is a write-only register and RDR04 is a read-only register. These registers are located in the same address, so the read value is different from the write value. Therefore, instructions that perform a read-modify-write (RMW) operation, such as the INC/DEC instruction, cannot be used.

Figure 32.4-4 Transmission and Reception Data registers 04 (RDR04 / TDR04)

■ Reception:

RDR04 is the register that contains reception data. The serial data signal transmitted to the SIN04 pin is converted in the shift register and stored there. When the data length is 7 bits, the uppermost bit (D7) contains 0. When reception is complete the data is stored in this register and the reception data full flag (SSR04: RDRF) is set to 1. If a reception interrupt request is enabled at this point, a reception interrupt request occurs.

Read RDR04 when the RDRF flag of the status register (SSR04) is 1. The RDRF flag is cleared automatically to 0 when RDR04 is read. Also the reception interrupt is cleared if it is enabled and no error has occurred.

Data in RDR04 is invalid when a reception error occurs (SSR04: PE, ORE, or FRE = 1).

■ Transmission:

When data to be transmitted is written to the transmission data register (TDR04) in transmission enable state, it is transferred to the transmission shift register, then converted to serial data, and transmitted from the serial data output terminal (SOT04 pin). If the data length is 7 bits, the uppermost bit (D7) is not sent.

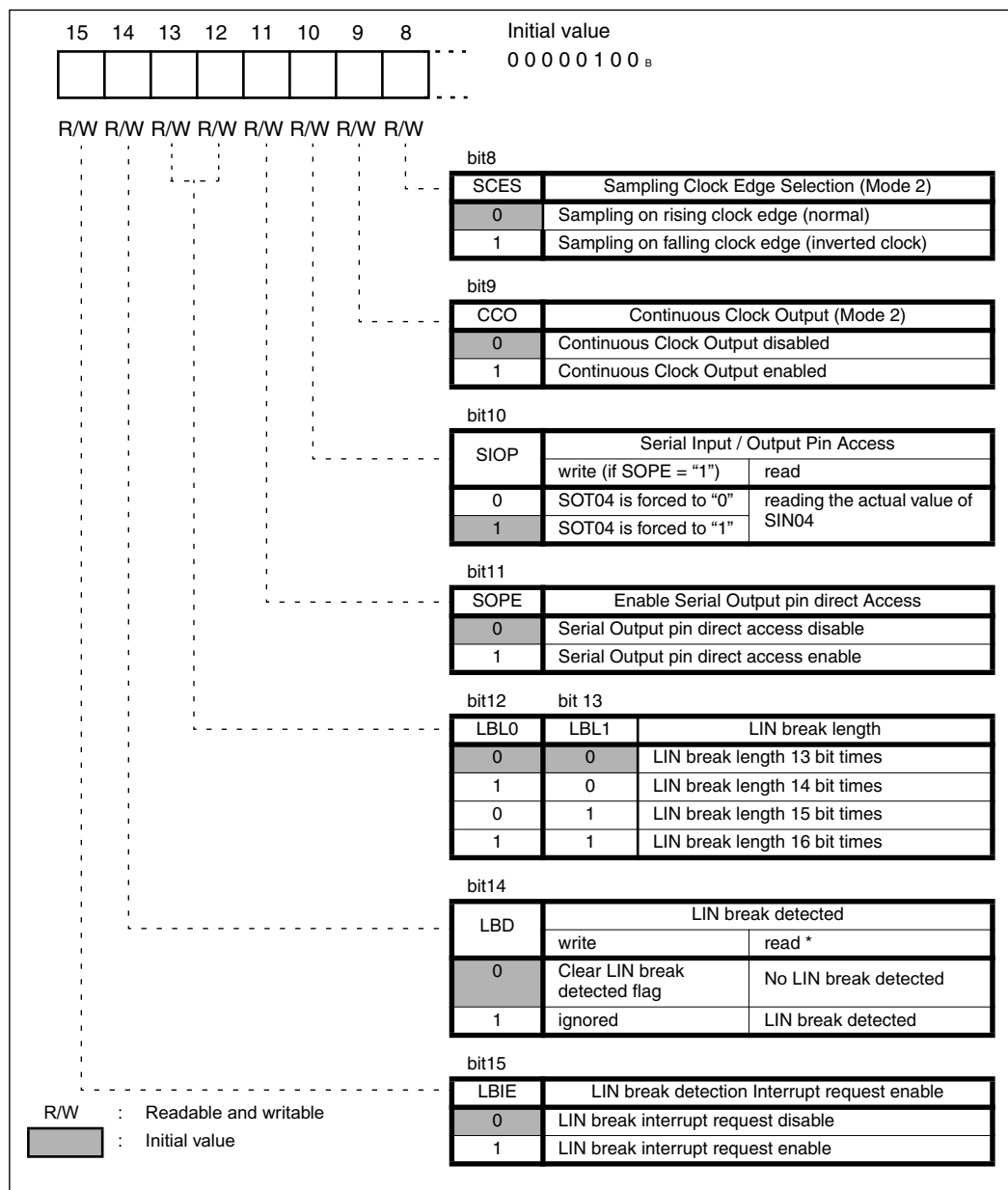
When transmission data is written to this register, the transmission data empty flag (SSR04: TDRE) is cleared to 0. When transfer to the transmission shift register is complete, the flag is set to 1. When the TDRE flag is 1, the next part of transmission data can be written. If output transmission interrupt requests have been enabled,

a transmission interrupt request is generated. Write the next part of transmission data when a transmission interrupt is generated or the TDRE flag is 1.

32.4.5 Extended Status/Control Register (ESCR04)

This register provides several LIN functions, direct access to the SIN04 and SOT04 pin and setting for USART synchronous clock mode.

Figure 32.4-5 Configuration of the Extended Status/Control Register (ESCR04)



* see table 32.4-6 for RMW access!

Table 32.4-6 Function of each bit of the Extended Status/Control Register (ESCR4)

Bit name		Function
bit15	LBIE: LIN break detection interrupt request enable bit	This bit enables a reception interrupt request, if a LIN break was detected.
bit14	LBD: LIN break detected flag	This flag goes 1 if a LIN break was detected. Writing a 0 to it clears this flag and the corresponding interrupt request, if it is enabled. Note: RMW instructions always return "1". In this case, the value "1" does not indicate a LIN-Break.
bit13 bit12	LBL1/0: LIN break length selection	These two bits determine how many serial bit times the LIN break is generated by USART. Receiving a LIN break is always fixed to 13 bit times.
bit11	SOPE: Serial Output pin direct access enable*	Setting this bit to 1 enables the direct write to the SOT04 pin, if SOE = 1 (SMR04).*
bit10	SIOP: Serial Input/Output Pin direct access*	Normal read instructions always return the actual value of the SIN04 pin. Writing to it sets the bit value to the SOT04 pin, if SOPE = 1. During a Read-Modify-Write instruction the bit returns the SOT04 value in the read cycle.*
bit9	CCO: Continuous Clock Output enable bit	This bit enables a continuous serial clock at the SCK04 pin if USART operates in master mode 2 (synchronous) and the SCK04 pin is configured as a clock output.
bit8	SCES: Serial clock edge selection bit	This bit inverts the internal serial clock in mode 2 (synchronous) and the output clock signal, if USART operates in master mode 2 (synchronous) and the SCK04 pin is configured as a clock output. In slave mode 2 the sampling time turns from rising edge to falling edge.

* see table 32.4-7 for SOPE and SIOP interaction

Table 32.4-7 * Description of the interaction of SOPE and SIOP:

SOPE	SIOP	Writing to SIOP	Reading from SIOP
0	R/W	has no effect on the SOT4 pin but holds the written value.	returns current value of SIN04
1	R/W	write "0" or "1" to SOT04	returns current value of SIN04
1	RMW		returns current value of SOT04 and writes it back

MB91460 Series

32.4.6 Extended Communication Control Register (ECCR04)

The extended communication control register provides bus idle recognition, interrupt settings, synchronous clock settings, and the LIN break generation.

Figure 32.4-6 Configuration of the Extended Communication Control Register (ECCR04)

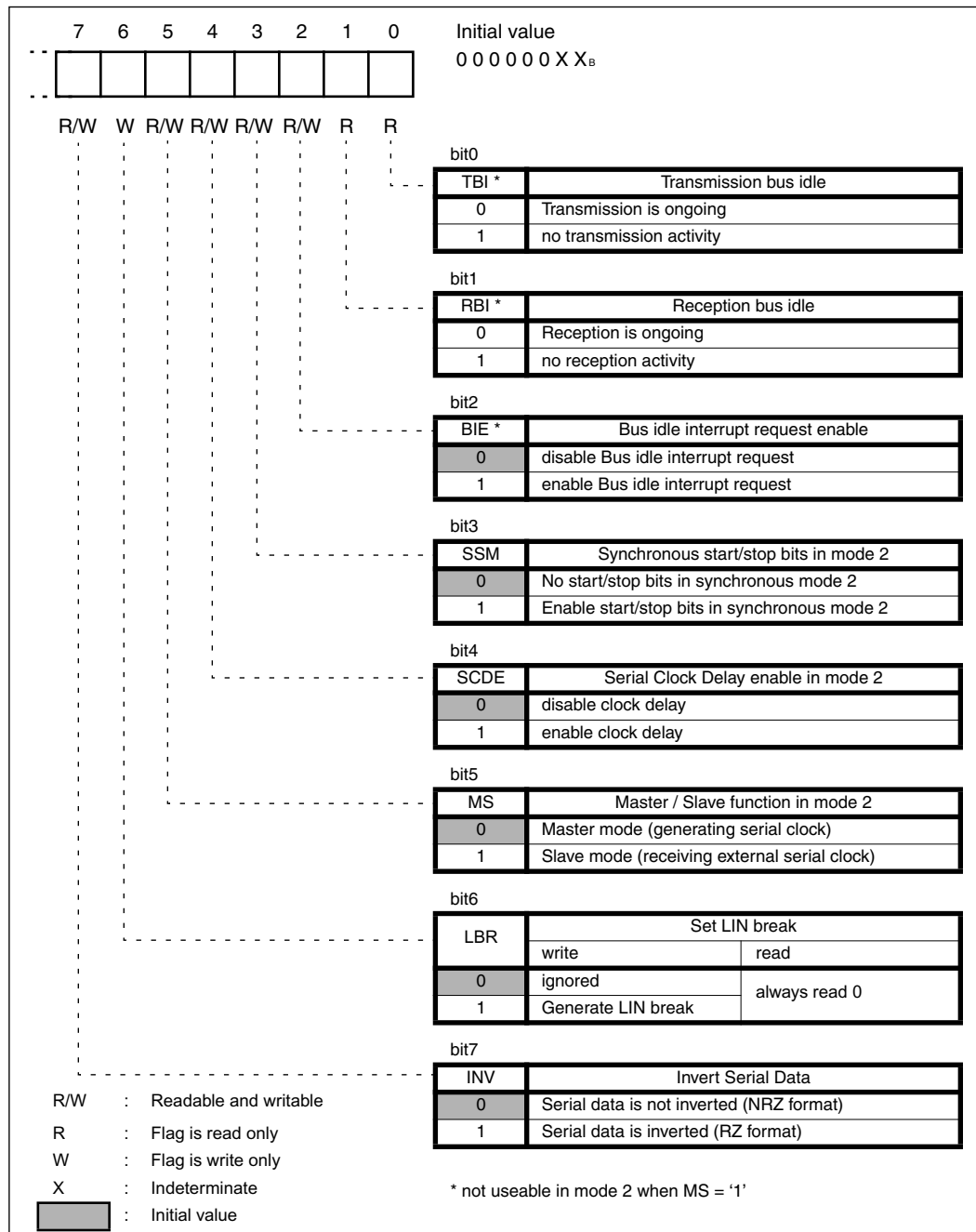


Table 32.4-8 Function of each bit of the Extended Communication Control Register (ECCR04)

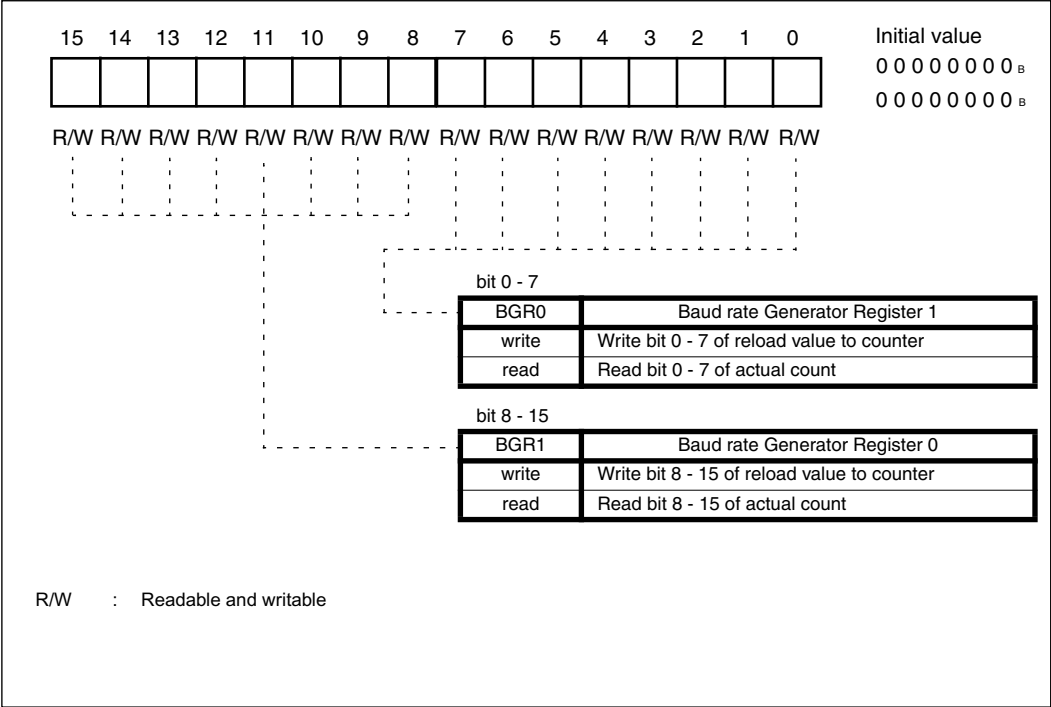
Bit name		Function
bit7	INV: Invert serial data	This bit inverts the serial data at SIN04 and SOT04 pin. SCK04 is not affected (see ESCR04: SCES). Writing "0": The serial data format is NRZ (default) Writing "1": The serial data is inverted (RZ format) RMW instructions do not affect this bit.
bit6	LBR: Set LIN break bit	Writing a 1 to this bit generates a LIN break of the length selected by the LBL0/1 bits of the ESCR04, if operation mode 0 or 3 is selected.
bit5	MS: Master/Slave mode selection bit	This bit selects master or slave mode of USART in synchronous mode 2. If master is selected USART generates the synchronous clock by itself. If slave mode is selected USART receives external serial clock. <Caution> If slave mode is selected, the clock source must be external and set to "One-to-One" (SMR04: SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	If this bit is set, the serial output clock is delayed by 1 CLKP cycle (or half of its period in SPI-compliance). This only applies, if USART operates in master mode 2.
bit3	SSM: Start/Stop bit mode enable	This bit adds start and stop bits to the synchronous data format in operation mode 2. It is ignored in mode 0, 1, and 3. Setting SSM bit is only possible if operation mode is set to 2. (SMR04, MD bits)
bit2	BIE: Bus idle interrupt request enable	This bit enables a reception interrupt request, if there is neither reception nor transmission ongoing (RBI = 1, TBI = 1). Note: Do not use BIE in mode 2 when MS = 1.
bit1	RBI: Reception bus idle flag	This flag is "1" if there is no reception activity on the SIN04 pin. Note: Do not use this flag in mode 2 when MS = 1.
bit0	TBI: Transmission bus idle flag	This flag is "1" if there is no transmission activity on the SOT04 pin. Note: Do not use this flag in mode 2 when MS = 1.

Note: The RBI and TBI flags can be cleared through an interrupt. Then the interrupt request is cleared, while the interrupt service routine is active.

32.4.7 Baud Rate / Reload Counter Register 0 and 1 (BGR04 / 14)

The baud rate / reload counter registers set the division ratio for the serial clock. Also the actual count of the transmission reload counter can be read.

Figure 32.4-7 Baudrate Reload Counter Register 0 and 1 (BGR04 / 14)



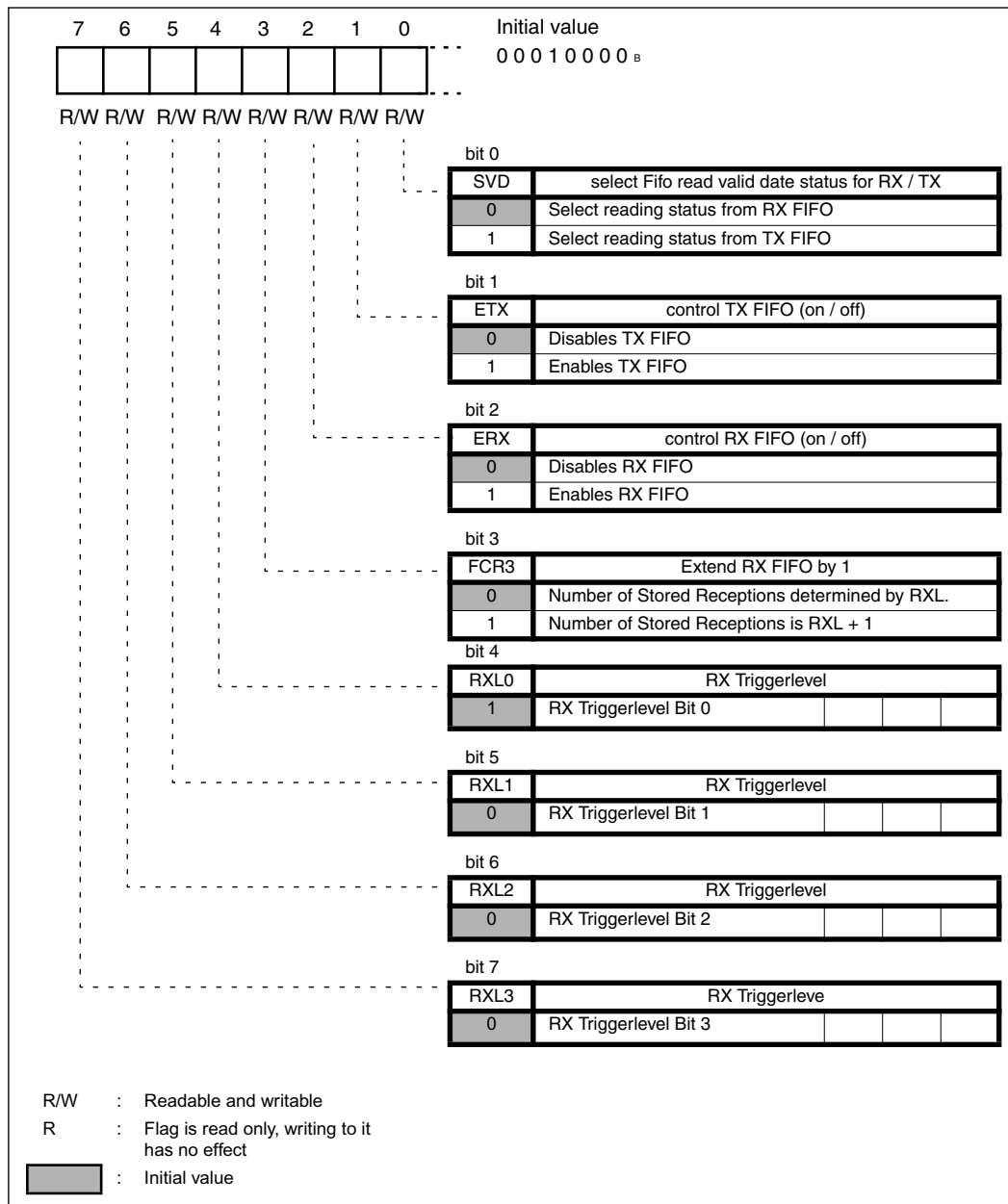
The Baud Rate / Reload Counter Registers determine the division ratio for the serial clock.

Both registers can be read or written via byte or word access.

Note: There are two independent reload counters available. One for transmission serial clock, the other one for reception serial clock.

32.4.8 FIFO Control Register (FCR04)

Figure 32.4-8 Configuration of FIFO control register



Note: Writing the FCR results in a reset of the FIFO buffer. E.g. a reconfiguration of the triggerlevels during a transmission or reception resets the FIFO buffer.

Table 32.4-9 Functions of each bit of fifo control register (FCR4)

Bit name		Function
bit 0	SVD: select Valid Data Fifo read	<ul style="list-style-type: none"> •If this bit is set to 0 the fifo status register shows the number of valid data from the RX fifo •If this bit is set to 1 the fifo status register shows the number of valid data from the TX fifo
bit 1	ETX: enable TX fifo	<ul style="list-style-type: none"> •If this bit is set to 0 the TX fifo is disabled / fifo data is cleared •If this bit is set to 1 the TX fifo is enabled.
bit 2	ERX: enable RX fifo	<ul style="list-style-type: none"> •If this bit is set to 0 the RX fifo is disabled / fifo data is cleared •If this bit is set to 1 the RX fifo is enabled.
bit 3	FCR3	<ul style="list-style-type: none"> •If this bit is set to 0 the RX fifo stores upto 15 receptions. •If this bit is set to 1 the RX fifo stores upto 16 receptions.
bit 4	RXL0: RX Triggerlevel bit 0	•Set the triggerlevel of RX Interrupt
bit 5	RXL1: RX Triggerlevel bit 1	•Set the triggerlevel of RX Interrupt
bit 6	RXL2: RX Triggerlevel bit 2	•Set the triggerlevel of RX Interrupt
bit 7	RXL3: RX Triggerlevel bit 3	•Set the triggerlevel of RX Interrupt

Note: The RX triggerlevel sets the reception FIFO level where the reception interrupt is activated. E.g. if the triggerlevel is at its default value of RXL[3:0]=0001, the interrupt is activated immediately. If the triggerlevel is set to RXL[3:0]=1111, the interrupt is activated if 15 receptions are stored in the FIFO.
 In general: a reception interrupt is triggered if FSR[4:0] >= FCR[7:4].

The devices listed below are not affected by FCR3 setting.

For these devices, writing 1 is not possible.(FCR3 = 0)

MB91F464A

MB91F465D

MB91F465X

MB91F467B

MB91F467C

MB91F467D

MB91F467M

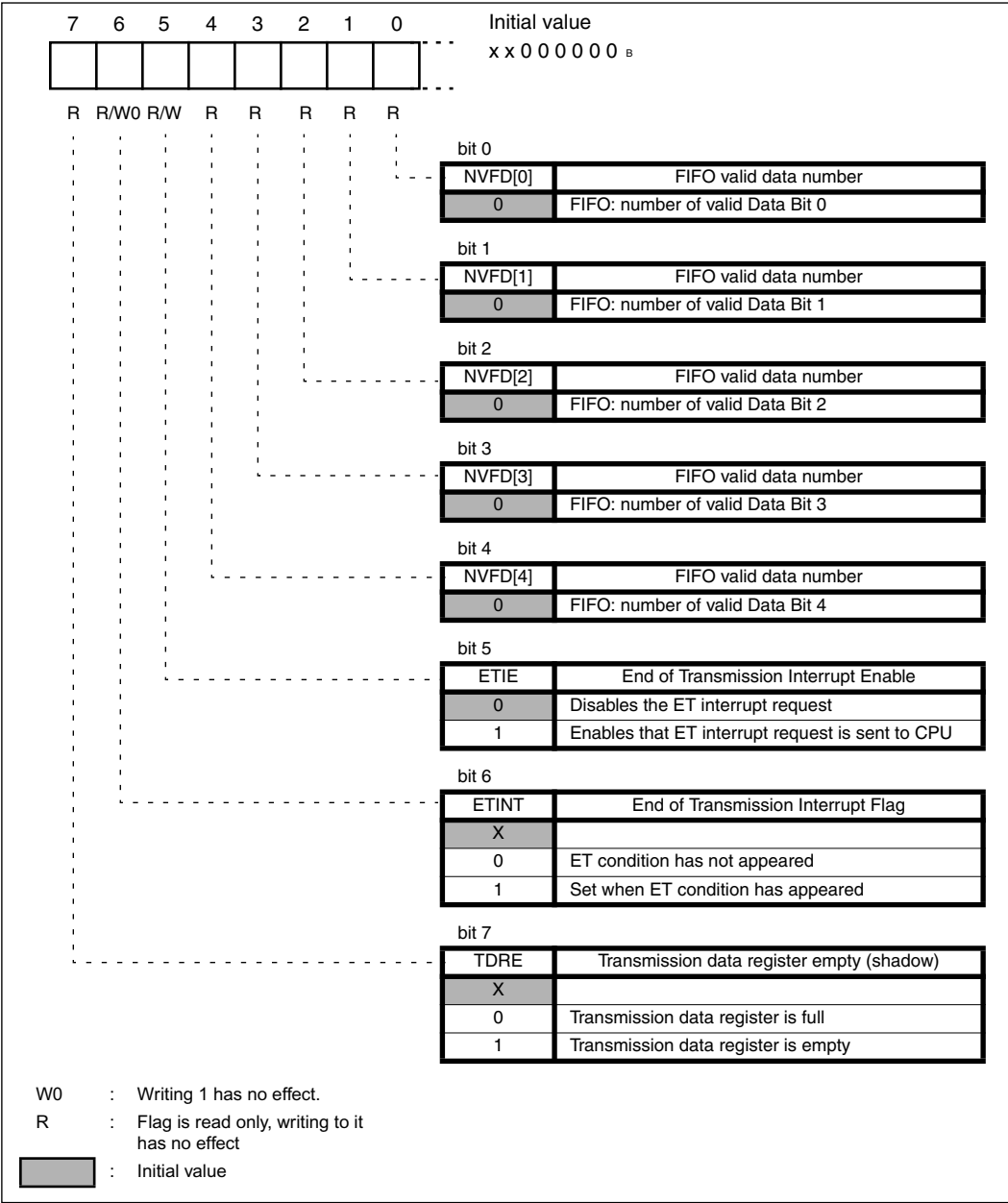
MB91F467R

MB91F469G

Note: When the configuration of the FIFO is changed (write access to FCR register) the FIFO becomes initialized.

32.4.9 FIFO Status Register (FSR04)

Figure 32.4-9 Configuration of FIFO status register



Note: The FSR04[4:0] FIFO valid data bits (NVFD[4:0]) indicate the number of stored receptions (SVD=0) or pending transmissions (SVD=1) in the FIFO buffer.

Table 32.4-10 Functions of each bit of FIFO status Register

Bit name		Function
bit [4:0]	NVFD[4:0]: Number of Valid FIFO Data.	<ul style="list-style-type: none"> Shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit. These bits indicate the number of stored receptions (SVD=0) or pending transmissions (SVD=1) in the FIFO buffer. If no FIFO is installed, these bits return 0x00.
bit 5	ETIE: End of Transmission Interrupt Enable ¹	<ul style="list-style-type: none"> ETIE = 1 enables that the EoT interrupt request is sent to the CPU when ETINT is set. ETIE = 0 (default) disables the EoT interrupt request. This bit is cleared by software reset (RST) and can be written and read by CPU.
bit 6	ETINT: End of Transmission Interrupt Flag	<ul style="list-style-type: none"> This flag is set when the EoT condition has appeared: If no FIFO is installed, after the last bit of a transmission has been sent, If FIFO is installed, after the last bit of a transmission has been sent and the FIFO is empty. This flag is cleared by software reset (RST) or by writing 0. Writing 1 has no effect. Read - modify - write access always reads 1.
bit 7	TDRE: Transmission Data Register Empty (shadow)	<ul style="list-style-type: none"> This is a read-only shadow of TDRE flag. Interrupt routines can determine the interrupt source (TDRE or EoT) by just reading the FSR register.

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P. On other devices, these bits are read-only and read '0'.

32.5. USART Interrupts

The USART uses both reception and transmission interrupts. An interrupt request can be generated for either of the following causes:

- Receive data is set in the Reception Data Register (RDR04), or a reception error occurs.
- Transmission data is transferred from the Transmission Data Register (TDR04) to the transmission shift register.
- Transmission data has been sent completely (End of Transmission) ¹
- A LIN break is detected
- No bus activity (neither reception nor transmission)

32.5.1 USART Interrupts

Table 32.5-1 USART Interrupts

RX/ TX/ ICU	IRQ request flag	Flag Register	Operation mode				Interrupt cause	Interrupt cause enable bit	How to clear the Interrupt Request
			0	1	2	3			
Reception	RDRF	SSR	x	x	x	x	receive data is written to RDR (FIFO level reached)	SSR:RIE	Receive data is read
	ORE	SSR	x	x	x	x	Overrun error		"1" is written to clear rec. error bit (SCR:CRE)
	FRE	SSR	x	x	*1	x	Framing error		
	PE	SSR	x		*2		Parity error		
	LBD	ESCR	x			x	LIN synch break detected	ESCR: LBIE	"0" is written to ESCR:LBD
	TBI & RBI	ESCR	x	x		x	no bus activity	ECCR:BIE	Receive data / Send data
Transmission	TDRE	SSR or FSR *3	x	x	x	x	Empty transmission register	SSR:TIE	Transfer data is written
	EoT *4	FSR	x	x	x	x	End of transmission [and FIFO empty *5]	FSR:ETIE	"0" is written to FSR:ETINT
Input Capture Unit	ICP4	IPCP	x			x	1st falling edge of LIN synch field	IPCP:ICE	disable ICE temporary
	ICP4	IPCP	x			x	5th falling edge of LIN synch field	IPCP:ICE	disable ICE

1. Only available if ECCR04/SSM = 1

2. Only available if ECCR04/SSM = 1

3. FSR:TDRE is a read-only mirror of the SSR:TDRE bit

4. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

5. if FIFO is installed

X: Used

1. End of Transmission interrupt is only available on MB91FV460B, MB91F467E and MB91F467P

32.5.2 Reception Interrupt

If one of the following events occur in reception mode, the corresponding flag of the Serial Status Register (SSR04) is set to '1':

- - Data reception is complete, i. e. the received data was transferred from the serial input shift register to the Reception Data Register (RDR04) and data can be read: **RDRF** (if FIFO is enabled, trigger level is reached)
- - Overrun error, i. e. RDRF = 1 and RDR04 was not read by the CPU: **ORE**
- - Framing error, i. e. a stop bit was expected, but a '0'-bit was received: **FRE**
- - Parity error, i. e. a wrong parity bit was detected: **PE**

If at least one of these flags above go '1' and the reception interrupt request is enabled (SSR04: RIE = 1), a reception interrupt request is generated.

If the Reception Data Register (RDR04) is read, the RDRF flag is automatically cleared to '0'. Note that this is the only way to reset the RDRF flag. The error flags are cleared to '0', if a '1' is written to the Clear Reception Error (CRE) flag bit of the Serial Control Register (SCR04). The RDR04 contains only valid data if the RDRF flag is '1' and no error bits are set.

Note that the CRE flag is 'write only' and by writing a '1' to it, it is internally held to '1' for one CLKP cycle.

32.5.3 Transmission Interrupt

If transmission data is transferred from the Transmission Data Register (TDR04) to the transfer shift register (this happens, if the shift register (or FIFO) is empty and transmission data exists), the Transmission Data Register Empty flag (TDRE) of the Serial Status Register (SSR04) is set to '1'. In this case an interrupt request is generated, if the Transmission Interrupt request Enable bit (TIE) of the SSR04 was set to '1' before.

Note, that the initial value of TDRE (after hardware or software reset) is '1'. So an interrupt is generated immediately then, if the TIE bit is set to '1'. Also note, that the only way to reset the TDRE flag is writing data to the Transmission Data Register (TDR04).

32.5.4 End of Transmission Interrupt (EoT)

The USART macros have been extended to generate an "End of Transmission" (EoT) interrupt after the last bit of a transmission has been sent. If EoT is enabled and there is no FIFO installed, the interrupt is generated after each transmission. If FIFO is installed, EoT appears after the transmission while the FIFO is empty.

The EoT interrupt cannot request a DMA transfer.

The EoT can be enabled and observed in the FSR (FIFO Status Register). Therefore, also USART modules which are not equipped with FIFO, have the FIFO Status Register.

32.5.5 LIN Synchronization Break Interrupt

This paragraph is only relevant, if USART operates in mode 0 or 3 as a LIN slave.

If the bus (serial input) goes '0' (dominant) for more than 11 bit times, the LIN Break Detected (LBD) flag of the Extended Status/Control Register (ESCR04) is set to '1'. Note, that in this case after 9 bit times the reception error flags are set to '1', therefore the RIE flag has to be set to '0' or the RXE flag has to be set to '0', if only a LIN synch break detect is desired. In the other case a reception error interrupt would be generated first, and the interrupt handler routine has then to wait for LBD = 1.

The interrupt and the LBD flag are cleared after writing a '1' to the LBD flag. This makes sure, that the CPU has detected the LIN synch break, because of the following procedure of adjusting the serial clock to the LIN master.

32.5.6 LIN Synchronization Field Edge Detection Interrupts

This paragraph is only relevant, if USART operates in mode 0 or 3 as a LIN slave. After a LIN break detection the next falling edge of the reception bus is indicated by USART. Simultaneously an internal signal connected to the ICU is set to '1'. This signal is reset to '0' after the fifth falling edge of the LIN Synchronization Field. In both cases the ICU4 generates an interrupt request, if "both edge detection" and the ICU1/5 interrupt request

are enabled. The difference of the ICU4 counter values is the serial clock multiplied by 8. Dividing it by 8 results in the new detected and calculated baud rate for the dedicated reload counter. This value - 1 has then to be written to the Baud Rate Generator Registers (BGR1/0). There is no need to restart the reload counter, because it is automatically reset if a falling edge of a start bit is detected.

32.5.7 Bus Idle Interrupt

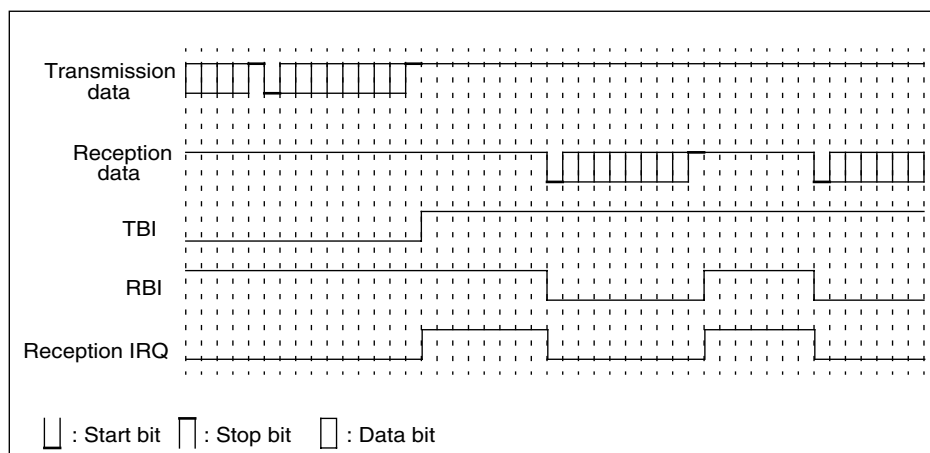
If there is no reception activity on the SIN04 pin, the RBI flag of the ECCR04 goes '1'. The TBI flag respectively goes '1', when no data is transmitted. If the Bus Idle Interrupt request Enable bit (BIE) of the ECCR04 is set and **both** bus idle flags (TBI **and** RBI) are '1', an interrupt request is generated.

Note: The TBI flag goes also '0' if there is no bus activity, but a '0' is written to the SIOP bit, if SOPE is '1'.

Note: TBI and RBI cannot be used in mode 2 (synchronous communication).

Figure 32.5-1 illustrates how the bus idle interrupt is generated

Figure 32.5-1 Bus idle interrupt generation



32.5.8 Reception Interrupt Request Generation and Flag Set Timing

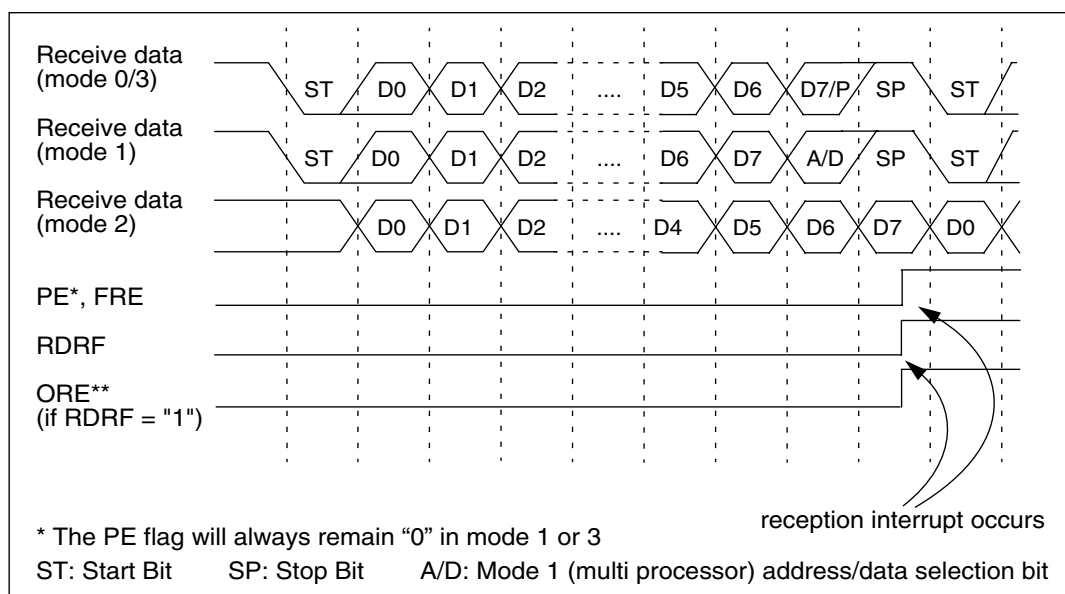
The following are the reception interrupt causes: Completion of reception (SSR04: RDRF) and occurrence of a reception error (SSR04: PE, ORE, or FRE).

■ Reception Interrupt Request Generation and Flag Set Timing

A reception interrupt request is generated, if the received data is complete (RDRF = 1) and the Reception Interrupt request Enable bit (RIE) of the Serial Status Register (SSR04) was set to '1'. This interrupt is generated if the first stop bit is detected in mode 0, 1, 2 (if SSM = 1), 3, or the last data bit was read in mode 2 (if SSM = 0).

Note: If a reception error has occurred, the Reception Data Register (RDR04) contains invalid data in each mode.

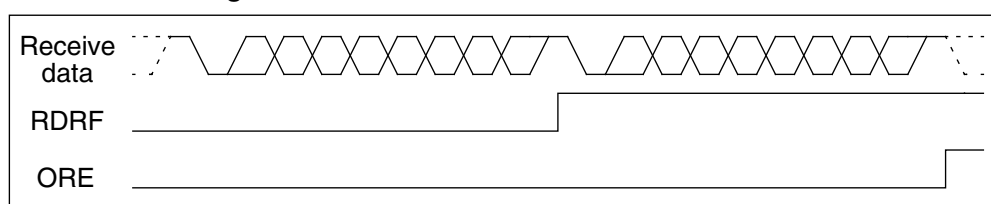
Figure 32.5-2 Reception operation and flag set timing



Note: The example in figure 32.5-2 does not show all possible reception options for mode 0 and 3. Here it is: '7p1' and '8N1' (p = 'E' [even] or 'O' [odd]), all in NRZ data format (ECCR04: INV = 0).

Note: **ORE only occurs, if the reception data is not read by the CPU (RDRF = 1) and another data frame is read.

Figure 32.5-3 ORE set timing



32.5.9 Transmission Interrupt Generation and Flag Set Timing

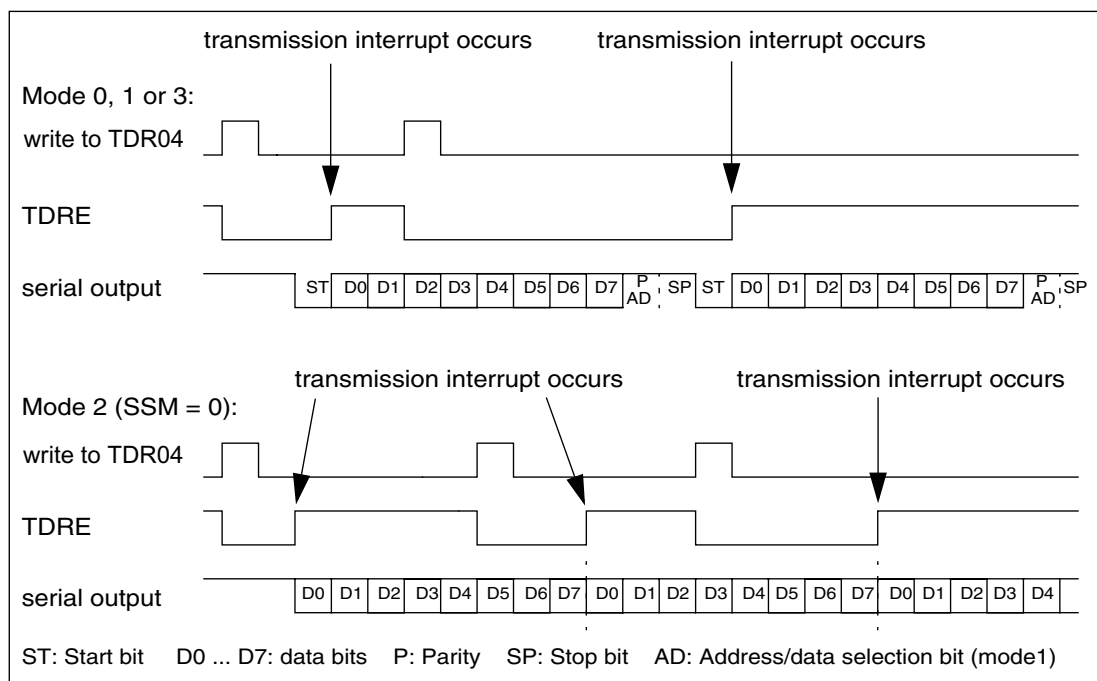
A transmission interrupt is generated when the next data to be sent is ready to be written to the output data register (TDR04).

■ Transmission Interrupt Generation and Flag Set Timing

A transmission interrupt request is generated, when the next data to be send is ready to be written to the Transmission Data Register (TDR04), i. e. the TDR04 is empty, and the transmission interrupt request is enabled by setting the Transmission Interrupt request Enable (TIE) bit of the Serial Status Register (SSR04) to '1'.

The Transmission Data Register Empty flag (TDRE) of the SSR04 indicates an empty TDR04. Because the TDRE flag is "read only", it only can be cleared by writing data into TDR04.

The following figure demonstrates the transmission operation and flag set timing for the four modes of USART.

Figure 32.5-4 Transmission operation and flag set timing

Note: The example in figure 32.5-4 does not show all possible transmission options for mode 0. Here it is: "8p1" (p = "E" [even] or "O" [odd]), ECCR04: INV = 0. Parity is not provided in mode 3 or 2, if SSM = 0.

■ Transmission Interrupt Request Generation Timing

If the TDRE flag is set to 1 when a transmission interrupt request is enabled (SSR04: TIE=1) a transmission interrupt request is generated.

A transmission completion interrupt request is generated immediately after the transmission interrupt request is enabled (TIE=1) because the TDRE flag is set to 1 as its initial value. TDRE is a read-only flag that can be cleared only by writing new data to the output data register (TDR04). Carefully specify the transmission interrupt enable timing.

32.6. USART Baud Rates

One of the following can be selected for the USART serial clock source:

- Dedicated baud rate generator (Reload Counter)
- External clock as it is (clock input to the SCK04 pin)
- External clock connected to the baud rate generator (Reload Counter)

32.6.1 USART Baud Rate Selection

The baud rate selection circuit is designed as shown below. One of the following three types of baud rates can be selected:

- Baud Rates Determined Using the Dedicated Baud Rate Generator (Reload Counter)

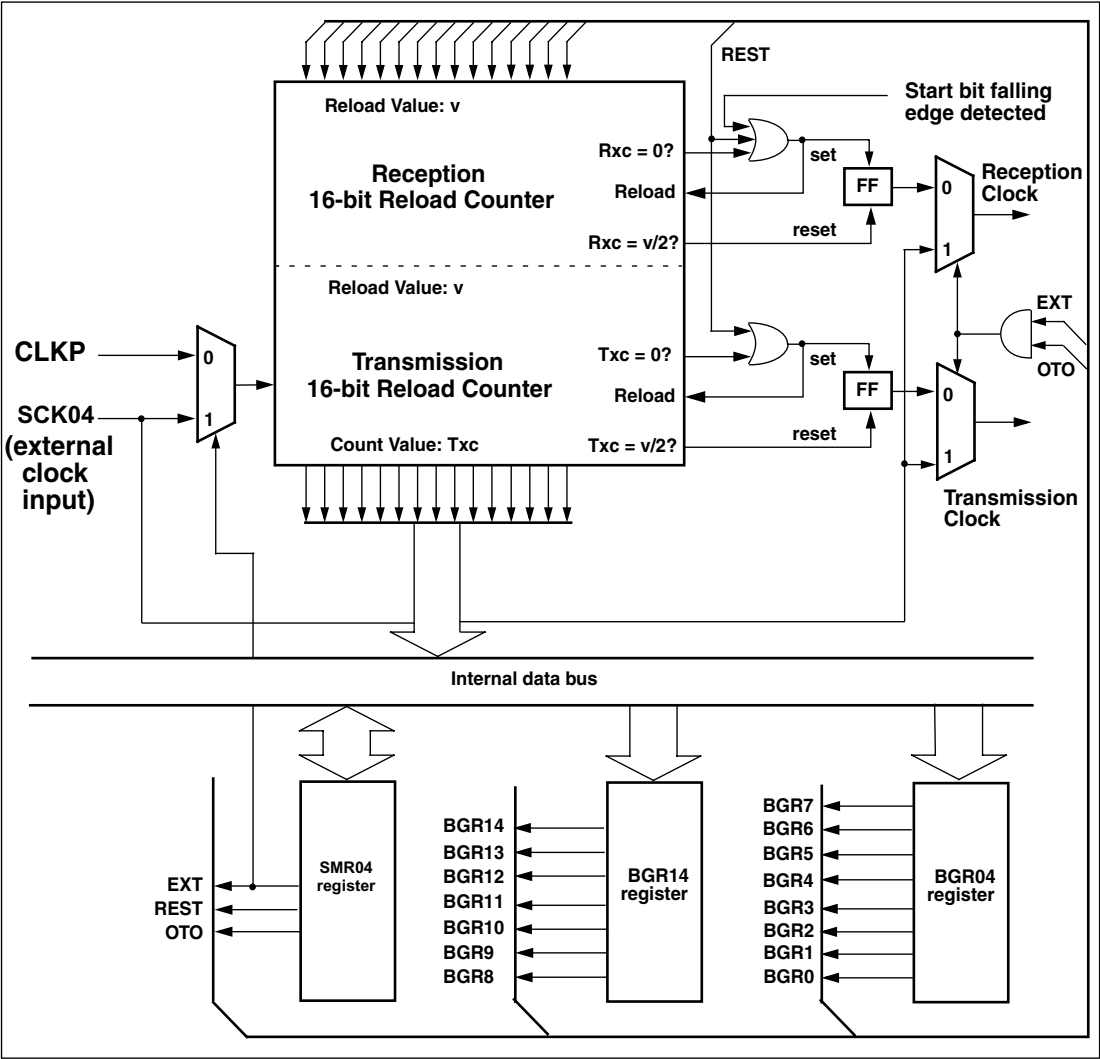
USART has two independent internal reload counters for transmission and reception serial clock. The baud rate can be selected via the 16-bit reload value determined by the Baud Rate Generator Register 0 and 1 (BGR0/1).

The reload counter divides the peripheral clock CLKP by the value set in the Baud Rate Generator Register 0 and 1.
- Baud Rates determined using external clock (one-to-one mode)

The clock input from USART clock pulse input pins (SCK04) is used as it is (synchronous). Any baud rate less than the peripheral clock divided by 4 and is divisible can be set externally
- Baud Rates determined using the dedicated baud rate generator with external clock

An external clock source can also be connected internally to the reload counter. In this mode it is used instead of the internal peripheral clock. This was designed to use quartz oscillators with special frequencies and having the possibility to divide them.

Figure 32.6-1 Baud rate selection circuit (reload counter)



32.6.2 Setting the Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ Calculating the baud rate

The both 15-bit Reload Counters are programmed by the Baud Rate Generator Registers 1 and 0 (BGR14, 04). The following calculation formula should be used to set the wanted baud rate:

Reload Value:

$$v = [F / b] - 1 ,$$

where F is the resource clock (CLKP), b the baud rate and $[]$ gaussian brackets (mathematical rounding function).

■ Example of Calculation

If CLKP is 16MHz and the desired baud rate is 19200 baud then the reload value v is:

$$v = [16 \cdot 10^6 / 19200] - 1 = \mathbf{832}$$

The exact baud rate can then be recalculated: $b_{exact} = F / (v + 1)$, here it is: $16 \cdot 10^6 / 833 = 19207.6831$

Note: Setting the reload value to 0 stops the reload counter.

Note: The minimum recommended division ratio is 4 (i.e. reload value is 3) due to RX oversampling filter in asynchronous communication modes (mode 0,1 and 3).

■ Suggested Division Ratios for different machine speeds and baud rates

The following settings are suggested for different CLKP speeds and baud rates:

Table 32.6-1 Suggested Baud Rates and reload values at different CLKP speeds.

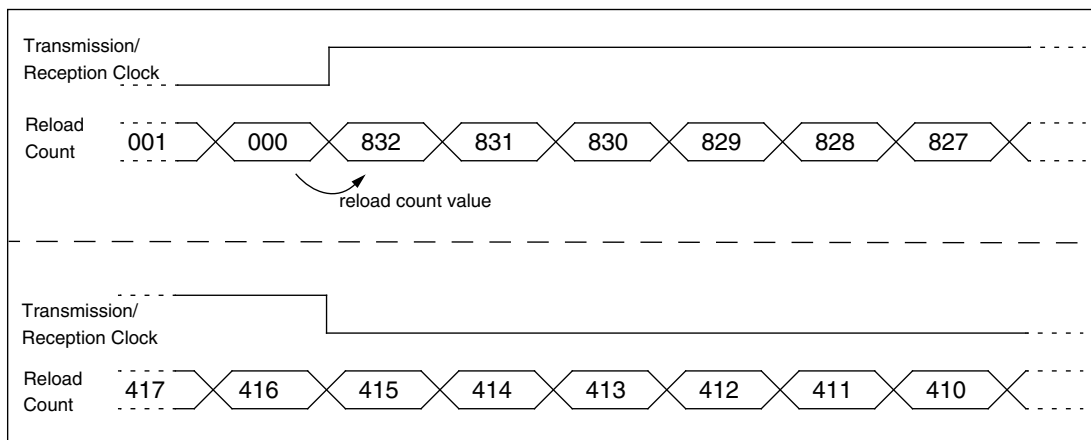
Baud rate	16MHz		24MHz		32MHz		40MHz		48MHz		50MHz	
	value	% dev.	value	% dev.	value	% dev.	value	% dev.	value	% dev.	value	% dev.
2M	7	0	11	0	15	0	19	0	23	0	24	0
1M	15	0	23	0	31	0	39	0	47	0	49	0
500000	31	0	47	0	63	0	79	0	95	0	99	0
460800	-	-	51	-0.16	-	-	86	0.22	103	-0.16	-	-
250000	63	0	95	0	127	0	159	0	191	0	199	0
230400	-	-	103	-0.16	138	0.08	173	0.22	207	-0.16	216	<0.01
153600	103	-0.16	155	-0.16	207	-0.16	259	-0.16	312	0.16	325	0.15
125000	127	0	191	0	255	0	319	0	383	0	399	0
115200	138	0.08	207	-0.16	278	0.08	346	-0.06	416	0.08	433	<0.01
76800	207	-0.16	311	-0.16	416	0.08	520	0.03	624	0	650	<0.01
57600	277	0.08	416	0.08	555	0.08	693	-0.06	832	-0.04	867	<0.01
38400	416	0.08	624	0	832	-0.04	1041	0.03	1249	0	1301	<0.01
28800	554	-0.01	832	-0.03	1110	-0.01	1388	<0.01	1666	0.02	1735	<0.01
19200	832	-0.03	1249	0	1666	0.02	2082	-0.02	2499	0	2603	<0.01
10417	1535	<0.01	2303	<0.01	3071	<0.01	3839	<0.01	4607	<0.01	4799	<0.01
9600	1666	0.02	2499	0	3332	0.01	4166	<0.01	4999	0	5207	<0.01
7200	2221	<0.01	3332	<0.01	4443	-0.01	5555	<0.01	6666	<0.01	6943	<0.01
4800	3332	<0.01	4999	0	6666	<0.01	8332	<0.01	9999	0	10416	<0.01
2400	6666	<0.01	9999	0	13332	<0.01	16666	<0.01	19999	0	20832	<0.01
1200	13332	<0.01	19999	0	26666	<0.01	33332	<0.01	39999	0	41666	<0.01
600	26666	<0.01	39999	0	53332	<0.01	-	-	-	-	-	-
300	53332	<-0.01	-	-	-	-	-	-	-	-	-	-

MB91460 Series

■ Counting Example

Assume the reload value is 832. The figure 32.6-2 demonstrates the behavior of the both Reload Counters:

Figure 32.6-2 Counting example of the reload counters



Note: The falling edge of the Serial Clock Signal always occurs after $\lfloor (v + 1) / 2 \rfloor$.

32.6.3 Restarting the Reload Counter

The Reload Counter can be restarted because of the following reasons:

Transmission and Reception Reload Counter:

- Global MCU Reset
- USART programmable clear (SMR04:UPCL bit)
- User programmable restart (SMR04: REST bit)

Reception Reload Counter:

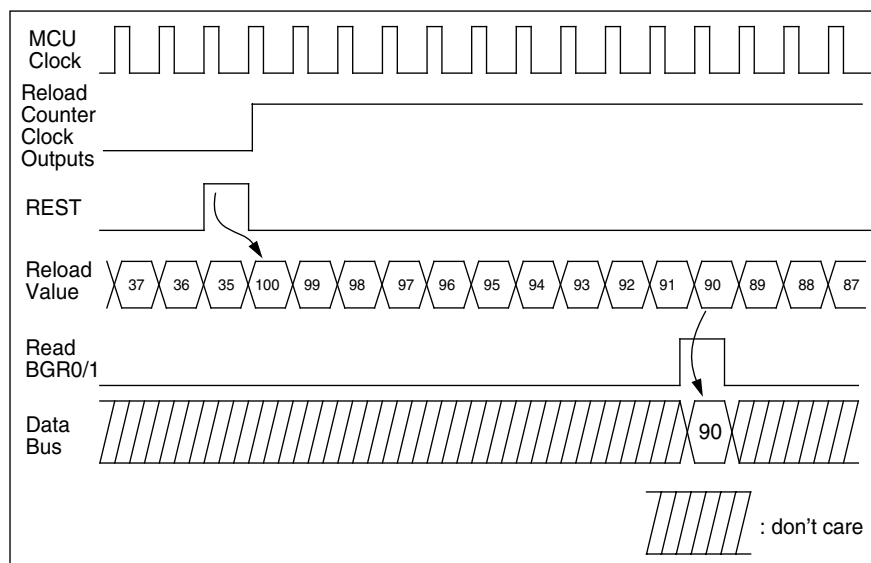
- Start bit falling edge detection in asynchronous mode

■ Programmable Restart

If the REST bit of the Serial Mode Register (SMR04) is set by the user, both Reload Counters are restarted at the next clock cycle. This feature is intended to use the Transmission Reload Counter as a small timer.

The following figure illustrates a possible usage of this feature (assume that the reload value is 100.)

Figure 32.6-3 Reload Counter Restart example



In this example the number of CLKP cycles (cyc) after REST is then:

$$cyc = v - c + 1 = 100 - 90 + 1 = 11,$$

where v is the reload value and c is the read counter value.

Note: If USART is reset by setting SMR04:UPCL, the Reload Counters will restart too.

■ Automatic Restart

In asynchronous UART mode if a falling edge of a start bit is detected the Reception Reload Counter is restarted. This is intended to synchronize the serial input shifter to the incoming serial data stream.

MB91460 Series

32.7. USART Operation

USART operates in operation mode 0 for normal bidirectional serial communication, in mode 2 and 3 in bidirectional communication as master or slave, and in mode 1 as master or slave in multiprocessor communication.

■ Operation of USART

- Operation modes

There are four USART operation modes: modes 0 to 3. As listed in table 32.7-1, an operation mode can be selected according to the inter-CPU connection method and data transfer mode.

Table 32.7-1 USART operation mode

Operation mode		Data length		Synchronization of mode	Length of stop bit	data bit direction*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first

** '+1' means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

Note: Mode 1 operation is supported both for master or slave operation of USART in a master-slave connection system. In Mode 3 the USART function is locked to 8N1-Format, LSB first.

If the mode is changed, USART cuts off all possible transmission or reception and awaits then new action.

■ Inter-CPU Connection Method

External Clock One-to-one connection (normal mode) and master-slave connection (multiprocessor mode) can be selected. For either connection method, the data length, whether to enable parity, and the synchronization method must be common to all CPUs. Select an operation mode as follows:

- In the one-to-one connection method, operation mode 0 or 2 must be used in the two CPUs. Select operation mode 0 for asynchronous transfer mode and operation mode 2 for synchronous transfer mode.

Note, that one CPU has to set to the master and one to the slave in synchronous mode 2.

- Select operation mode 1 for the master-slave connection method and use it either for the master or slave system.

■ Synchronization Methods

In asynchronous operation USART reception clock is automatically synchronized to the falling edge of a received start bit.

In synchronous mode the synchronization is performed either by the clock signal of the master device or by USART itself if operating as master.

■ Signal Mode

USART can treat data in non-return to zero (NRZ) and return to zero (RZ) format. For this option the ECCR: INV bit is provided.

■ Operation Enable Bit

USART controls both transmission and reception using the operation enable bit for transmission (SCR04: TXE) and reception (SCR04: RXE). If each of the operations is disabled, stop it as follows:

- If reception operation is disabled during reception (data is input to the reception shift register), finish frame reception and read the received data of the reception data register (RDR04). Then stop the reception operation.
- If the transmission operation is disabled during transmission (data is output from the transmission shift register), wait until there is no data in the transmission data register (TDR04) before stopping the transmission operation.

32.7.1 Operation in Asynchronous Mode (Op. Modes 0 and 1)

When USART is used in operation mode 0 (normal mode) or operation mode 1 (multiprocessor mode), the asynchronous transfer mode is selected.

■ Transfer data format

Each data transfer in the asynchronous mode operation begins with the start bit (low-level on bus) and ends with at least one stop bit (high-level). The direction of the bit stream (LSB first or MSB first) is determined by the BDS-Bit of the Serial Status Register (SSR04). The parity bit (if enabled) is always placed between the last data bit and the (first) stop bit.

In operation mode 0 the length of the data frame can be 7 or 8 bits, with or without parity, and 1 or 2 stop bits.

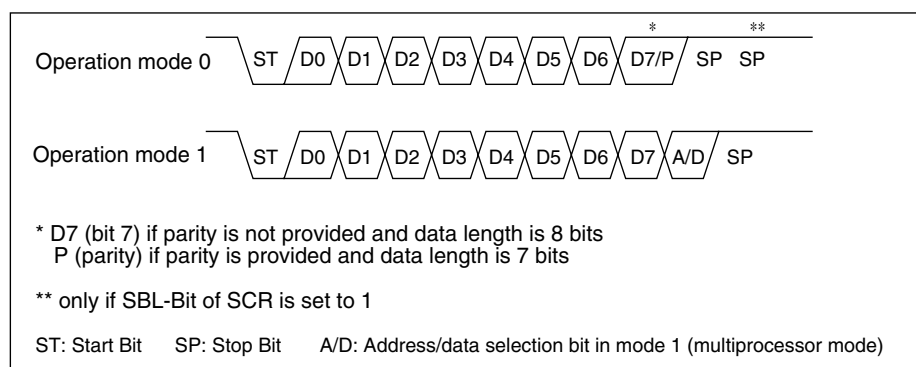
In operation mode 1 the length of the data frame can be 7 or 8 bits with a following address-/data-selection bit instead of a parity bit. 1 or 2 stop bits can be selected.

The calculation formula for the bit length of a transfer frame is:

$$\text{Length} = 1 + d + p + s$$

(d = number of data bits [7 or 8], p = parity [0 or 1], s = number of stop bits [1 or 2])

Figure 32.7-1 Transfer data format (operation modes 0 and 1)



Note: If BDS-Bit of the Serial Status Register (SSR04) is set to '1' (MSB first), the bit stream processes as: D7, D6, ..., D1, D0, (P).

During Reception both stop bits are detected, if selected. But the Reception data register full (RDRF) flag will go '1' at the first stop bit. The bus idle flag (RBI of ECCR04) goes '1' after the second stop bit if no further start bit is detected. (The second stop bit belongs to 'bus activity', although it is just mark level.)

■ Transmission Operation

If the Transmission Data Register Empty (TDRE) flag bit of the Serial Status Register (SSR04) is '1', transmission data is allowed to be written to the Transmission Data Register (TDR04). When data is written, the TDRE flag goes '0'. If the transmission operation is enabled by the TXE-Bit ('1') of the Serial Control

Register (SCR04), the data is written next to the transmission shift register and the transmission starts at the next clock cycle of the serial clock, beginning with the start bit. Thereby the TDRE flag goes '1', so that new data can be written to the TDR04.

If transmission interrupt request is enabled (TIE = 1), the interrupt request is generated by the TDRE flag. Note, that the initial value of the TDRE flag is '1', so that in this case if TIE is set to '1' an interrupt request will occur immediately.

■ Reception Operation

Reception operation is performed every time it is enabled by the Reception Enable (RXE) flag of the SCR04. If a start bit is detected, a data frame is received according to the format specified by the SCR04. By occurring errors, the corresponding error flags are set (PE, ORE, FRE). However after the reception of the data frame the data is transferred from the serial shift register to the Reception Data Register (RDR04) and the Receive Data Register Full (RDRF) flag of the SSR5 is set. The data then has to be read by the CPU. By doing so, the RDRF flag is cleared. If reception interrupt request is enabled (RIE = 1), the interrupt request is simply generated by the RDRF.

Note: Only when the RDRF flag is set and no errors have occurred the Reception Data Register (RDR04) contains valid data.

■ Stop Bit, Error Detection, and Parity

For transmission, 1 or 2 stop bits can be selected. During reception, if selected, both stop bits are checked, to set the reception bus idle (RBI) flag of ECCR04 correctly after the second stop bit.

In mode 0 parity, overrun, and framing errors can be detected.

In mode 1, overrun and framing errors can be detected. Parity is not provided.

By setting the Parity Enable (PEN) bit of the Serial Control Register (SCR04) the USART provides parity calculation (during transmission) and parity detection and check (during reception) in mode 0 (and mode 2 if the SSM bit of ECCR04 is set).

Even parity is set, if the P bit of SCR04 is cleared, odd parity if the flag is set. In mode 1, overrun and framing errors can be detected. Parity is not provided.

■ Signal mode NRZ and RZ

To set USART to the NRZ data format set the ECCR04:INV bit to 0 (initial value).

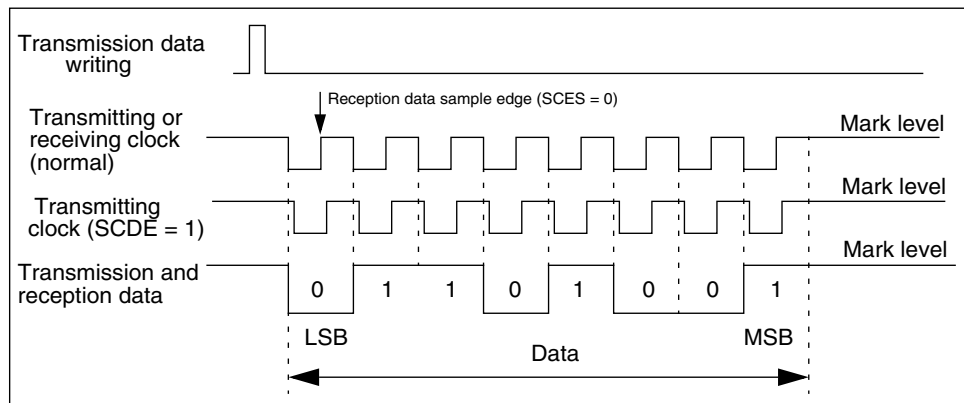
RZ data format is set, if the ECCR04:INV bit was set to 1.

32.7.2 Operation in Synchronous Mode (Operation Mode 2)

The clock synchronous transfer method is used for USART operation mode 2 (normal mode).

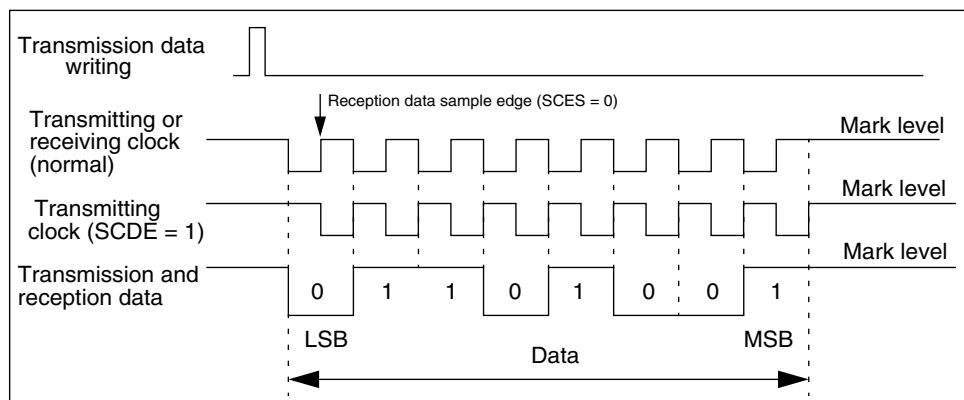
■ Transfer data format (standard synchronous)

In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR04) is 0. A special clock signal belongs to the data format in mode 2. The figure below illustrates the data format during a transmission in the synchronous operation mode

Figure 32.7-2 Transfer data format (operation mode 2).

■ Transfer data format

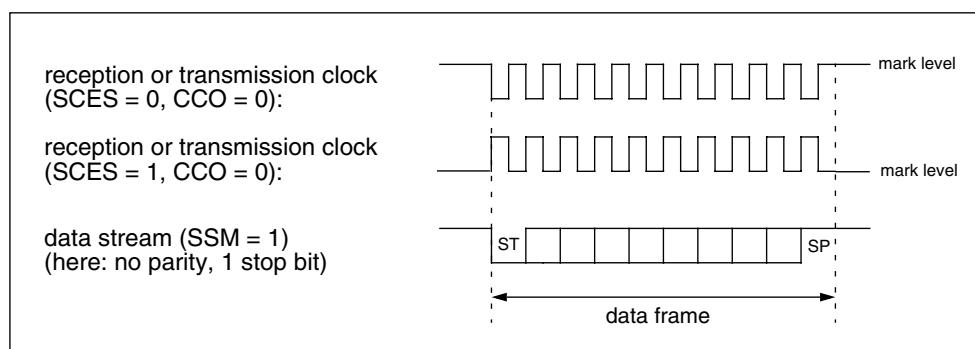
In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR04) is 0. A special clock signal belongs to the data format in mode 2. The figure below illustrates the data format during a transmission in the synchronous operation mode.

Figure 32.7-3 SPI Transfer data format (operation mode 2)

■ Clock inversion and start/stop bits in mode 2

If the SCES bit of the Extended Status/Control Register (ESCR04) is set the serial clock is inverted. Therefore in slave mode USART samples the data bits at the falling edge of the received serial clock. Note, that in master mode if SCES is set the clock signal's mark level is "0"¹. If the SSM04 bit of the Extended Communication Control Register (ECCR04) is set the data format gets additional start and stop bits like in asynchronous mode.

1. See the note about "Software reset of UART" on P. 803 if the mark level is "0".

Figure 32.7-4 Transfer data format with clock inversion

■ Clock Supply

In clock synchronous (normal) mode (I/O extended serial), the number of the transmission and reception bits has to be equal to the number of clock cycles. Note, that if start/stop bits communication is enabled, the number of clock cycles has to match with the quantity for the additional start and stop bit(s).

If the internal clock (dedicated reload counter) is selected, the data receiving synchronous clock is generated automatically if data is transmitted.

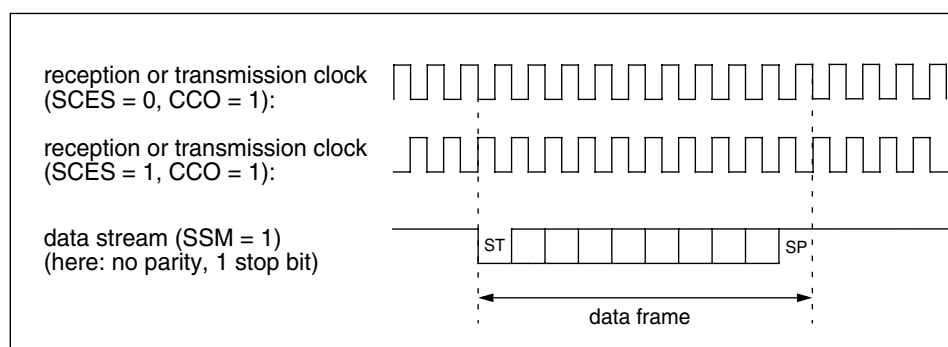
If external clock is selected, be sure, that the transmission side of the Transmission Data Register contains data and then clock cycles for each bit to sent have to be generated and supplied from outside. The mark level ('H') must be retained before transmission starts and after it is complete if SCES is "0".

Setting the SCDE bit of ECCR delays the transmitting clock signal by 1 CLKP cycle (or half a clock period in SPI). This will make sure, that the transmission data is valid and stable at any falling clock edge. (Necessary, if the receiving device samples the data at falling clock edge). This function is disabled when CCO is enabled.

If the Serial Clock Edge Select (SCES) bit of the ESCR is set, the USART's clock is inverted and thus samples the reception data at the falling clock edge. In this case, the sending device must make sure that the serial data is valid at the falling serial clock edge.

When both the SCES and the SCDE bit are set, data is stable at the rising clock edge, as in the case of SCES = SCDE = 0. However, the marker value for idle state is inverted (low).

If the CCO bit of the Extended Status/Control Register (ESCR) is set, the serial clock on the SCK pin in master mode is continuously clocked out. It is strongly recommended to use start and stop bits in this mode to signalize the receiver, when a data frame begins and when it stops. Figure 32.7-5 illustrates this.

Figure 32.7-5 Continuous clock output in mode 2

■ Data signal mode

NRZ data format is selected, if ECCR04: INV = 0, otherwise the signal mode for the serial data input and output pin is RZ.

■ Error Detection

If no Start/Stop bits are selected (ECCR04: SSM = 0) only overrun errors are detected.

■ Communication

For initialization of the synchronous slave mode, the following settings have to be done:

- Baud Rate Generator Registers (BGR0/1):
Set the desired reload value for the dedicated Baud Rate Reload Counter
- Serial Control Register (SCR04):
 - RXE, TXE: set both of these flags to '0'
 - PEN: no parity provided - Value: don't care
 - P, SBL, A/D: no parity, no stop bit(s), no Address/Data selection - Value: don't care
 - CL: automatically fixed to 8-bit data - Value: don't care
 - CRE: '1' (the error flag is cleared for initialization, possible transmission or reception will cut off)
- Serial Mode Control Register (SMR04):
 - MD1, MD0: '10b' (Mode 2)
 - SCKE: '1' for dedicated Baud Rate Reload Counter
'0' for external clock input
 - SOE: '1' for transmission and reception
'0' for reception only
- Serial Status Register (SSR04):
 - BDS: '0' for LSB first, '1' for MSB first
 - RIE: '1' if interrupts are used; '0' if not
 - TIE: '1' if interrupts are used; '0' if not
- Extended Communication Control Register (ECCR04):
 - SSM: '0' if no start/stop bits are desired (normal)
'1' for adding start/stop bits (special)
 - MS: '0' for master mode (USART generates the serial clock);
'1' for slave mode (USART receives serial clock from the master device)
- Serial Control Register (SCR04):
 - RXE, TXE: set one or both of these control bits to '1' to begin communication.

To start the communication, write data into the Transmission Data Register (TDR04).

To receive data, disable the Serial Output Enable (SOE) bit of the SMR04 and write dummy data to TDR04.

Note: Setting continuous clock and start-/stop-bit mode, duplex transfer is possible like in asynchronous modes.

32.7.3 Operation with LIN Function (Operation Mode 3)

USART can be used either for LIN-Master devices or LIN-Slave devices. For this LIN function a special mode (3) is provided. Setting the USART to mode 3, configure the data format to 8N1-LSB-first format.

■ USART as LIN master

In LIN master mode, the master determines the baud rate of the whole sub bus. Therefore, slave devices have to synchronize to the master and the desired baud rate remains fixed in master operation after initialization.

Writing a '1' into the LBR bit of the Extended Status/Communication Register (ECCR04) generates a 13 - 16 bit times low-level on the SOT04 pin, which is the LIN synchronization break and the start of a LIN message.

MB91460 Series

Thereby the TDRE flag of the Serial Status Register (SSR04) goes '0' and is reset to '1' after the break, and generates a transmission interrupt request for the CPU (if TIE of SSR04 is '1').

The length of the Synchronization break to be sent can be determined by the LBL1/0 bits of the ESCR04 as follows:

Table 32.7-2 LIN break length

LBL1	LBL0	Length of Break
0	0	13 Bit times
0	1	14 Bit times
1	0	15 Bit times
1	1	16 Bit times

The Synch Field can be sent as a simple 0x55-Byte after the LIN break. To prevent a transmission interrupt, the 0x55 can be written to the TDR04 just after writing the '1' to the LBR bit, although the TDRE flag is '0'. The internal transmission shifter waits until the LIN break has finished and shifts the TDR04 value out afterwards. In this case no interrupt is generated after the LIN break and before the start bit.

■ USART as LIN slave

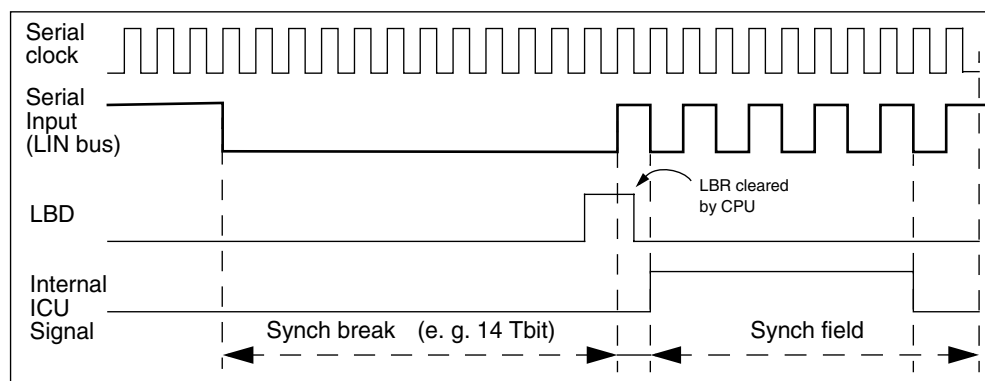
In LIN slave mode USART has to synchronize to the master's baud rate. If Reception is disabled (RXE = 0) but LIN break Interrupt request is enabled (LBIE = 1) USART will generate a reception interrupt request, if a synchronization break of the LIN master is detected, and indicates it with the LBD flag of the ESCR04. Writing a '0' to this flag clears the interrupt. The next step is to analyze the baud rate of the LIN master. The first falling edge of the Synch Field is detected by USART. The USART signals it then to the Input Capture Unit (ICU1/5) via a rising edge of an internal connection. The fifth falling edge resets the ICU signal. Therefore the ICU has to be configured for the LIN input capture (PFR14.4=1, EPFR14.4=1) and its interrupt requests have to be enabled (ICS4). The values of the ICU counter register after the first Interrupt (a) and after the second interrupt (b) yield the BGR value:

without timer overflow: $BGR \text{ value} = (b - a) / 8$,

with timer overflow: $BGR \text{ value} = (\max - b + a) / 8$,

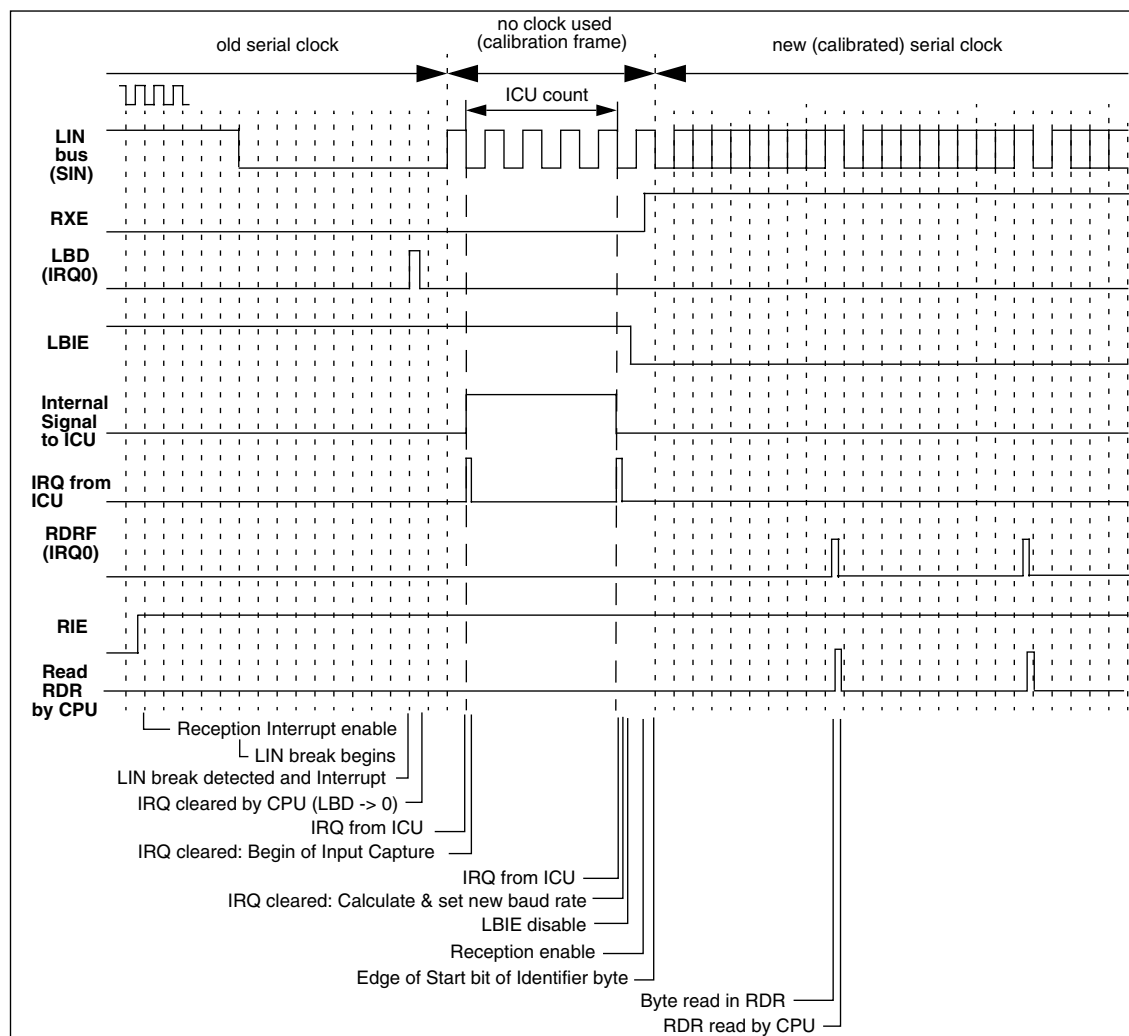
where max is the timer maximum value at which the overflow occurs.

The figure 32.7-6 shows a typical start of a LIN message frame and the behavior of the USART.

Figure 32.7-6 USART behavior as slave in LIN mode

■ LIN bus timing

Figure 32.7-7 LIN bus timing and USART signals



32.7.4 Direct Access to Serial Pins

USART allows the user to access directly the transmission pin (SOT04) or the reception pin (SIN04).

■ USART Direct Pin Access

The USART provides the ability for the user to directly access the serial input or output pin. The software can always monitor the incoming serial data by reading the SIOP bit of the ESCR04. If setting the Serial Output Pin direct access Enable (SOPE) bit of the ESCR04 the software can force the SOT04 pin to a desired value. Note, that this access is only possible, if the transmission shift register is empty (i. e. no transmission activity).

In LIN mode this function can be used for reading back the own transmission and is used for error handling if something is physically wrong with the single-wire LIN-bus.

Note: Write the desired value to the SIOP pin **before** enabling the output pin access, to prevent undesired peaks.

The peaks can occur because SIOP holds the last written value.

During a Read-Modify-Write operation the SIOP bit returns the actual value of the SOT04 pin in the read cycle instead the value of SIN04 during a normal read instruction.

32.7.5 Bidirectional Communication Function (Normal Mode)

In operation mode 0 or 2, normal serial bidirectional communication is available. Select operation mode 0 for asynchronous communication and operation mode 2 for synchronous communication.

■ Bidirectional Communication Function

The settings shown in figure 32.7-8 are required to operate USART in normal mode (operation mode 0 or 2).

Figure 32.7-8 Settings for USART operation mode 0 and 2

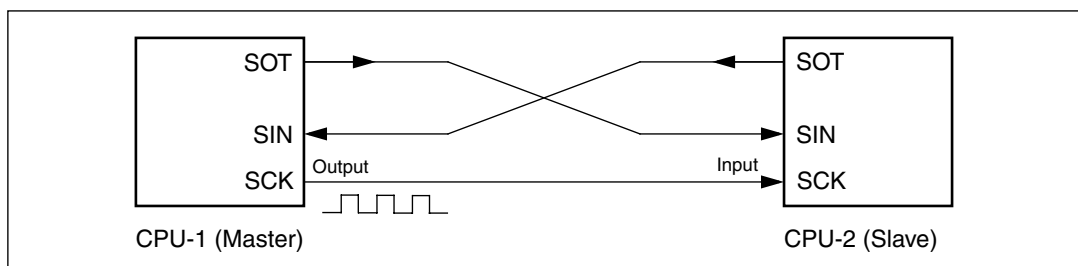
		bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04			PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 0			⊙	⊙	⊙	⊙	x	0	⊙	⊙	0	0	x	0	0	0	⊙	⊙
Mode 2			□	□	□	x	x	0	⊙	⊙	1	0	⊙	⊙	0	0	⊙	⊙
SSR04, TDR04/RDR04			PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 0			⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙								
Mode 2			□	⊙	□	⊙	⊙	⊙	⊙	⊙								
ESCR04,ECCR04			LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 0			x	x	x	x	x	x	x	x		x	x	x	x	x	x	x
Mode 2			x	x	x	x	x	x	⊙	⊙		x	⊙	⊙	⊙	x	x	x

⊙ Bit is used
 x Bit is not used
 0 / 1 Set bit to 0 / 1
 □ Bit is used if SSM = 1 (Synchronous start-/stop-bit mode)

■ Inter-CPU Connection

As shown in figure 32.7-9, interconnect two CPUs in USART mode 2

Figure 32.7-9 Connection example of USART mode 2 bidirectional communication



32.7.6 Master-Slave Communication Function (Multiprocessor Mode)

USART communication with multiple CPUs connected in master-slave mode is available for both master or slave systems.

■ Master-slave Communication Function

The settings shown in figure 32.7-10 are required to operate USART in multiprocessor mode (operation mode 1).

Figure 32.7-10 Settings for USART operation mode 1

		bit															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04		PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 1		x	x	⊙	⊙	⊙	0	⊙	⊙	0	1	x	0	0	0	1	⊙

SSR04, TDR04/RDR04		PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 1		x	⊙	⊙	⊙	⊙	⊙	⊙	⊙								

⊙ Bit is used

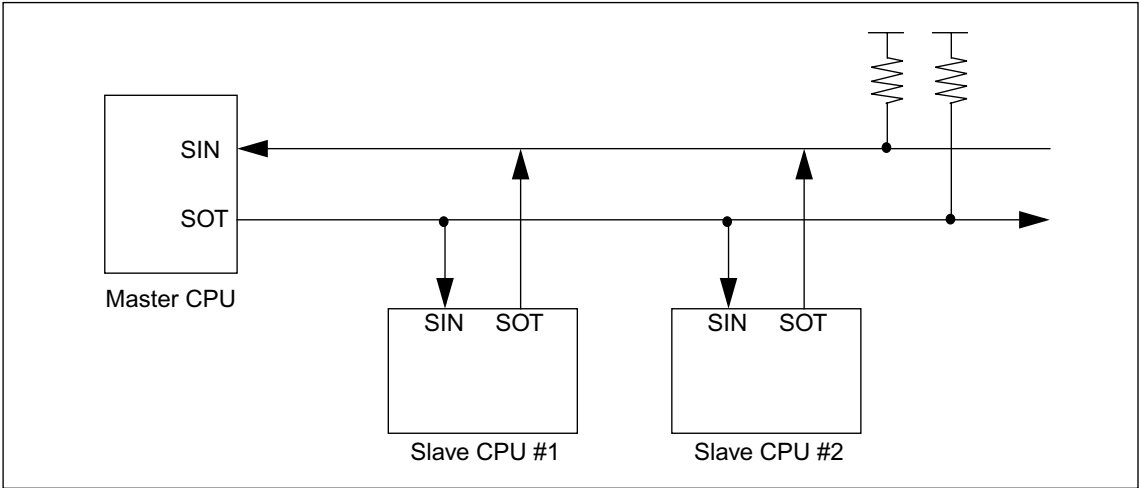
x Bit is not used

0 / 1 Set bit to 0 / 1

■ Inter-CPU Connection

As shown in figure 32.7-11, a communication system consists of one master CPU and multiple slave CPUs connected to two communication lines. USART can be used for the master or slave CPU.

Figure 32.7-11 Connection example of USART master-slave communication



MB91460 Series**■ Function Selection**

Select the operation mode and data transfer mode for master-slave communication as shown in table 32.7-3.

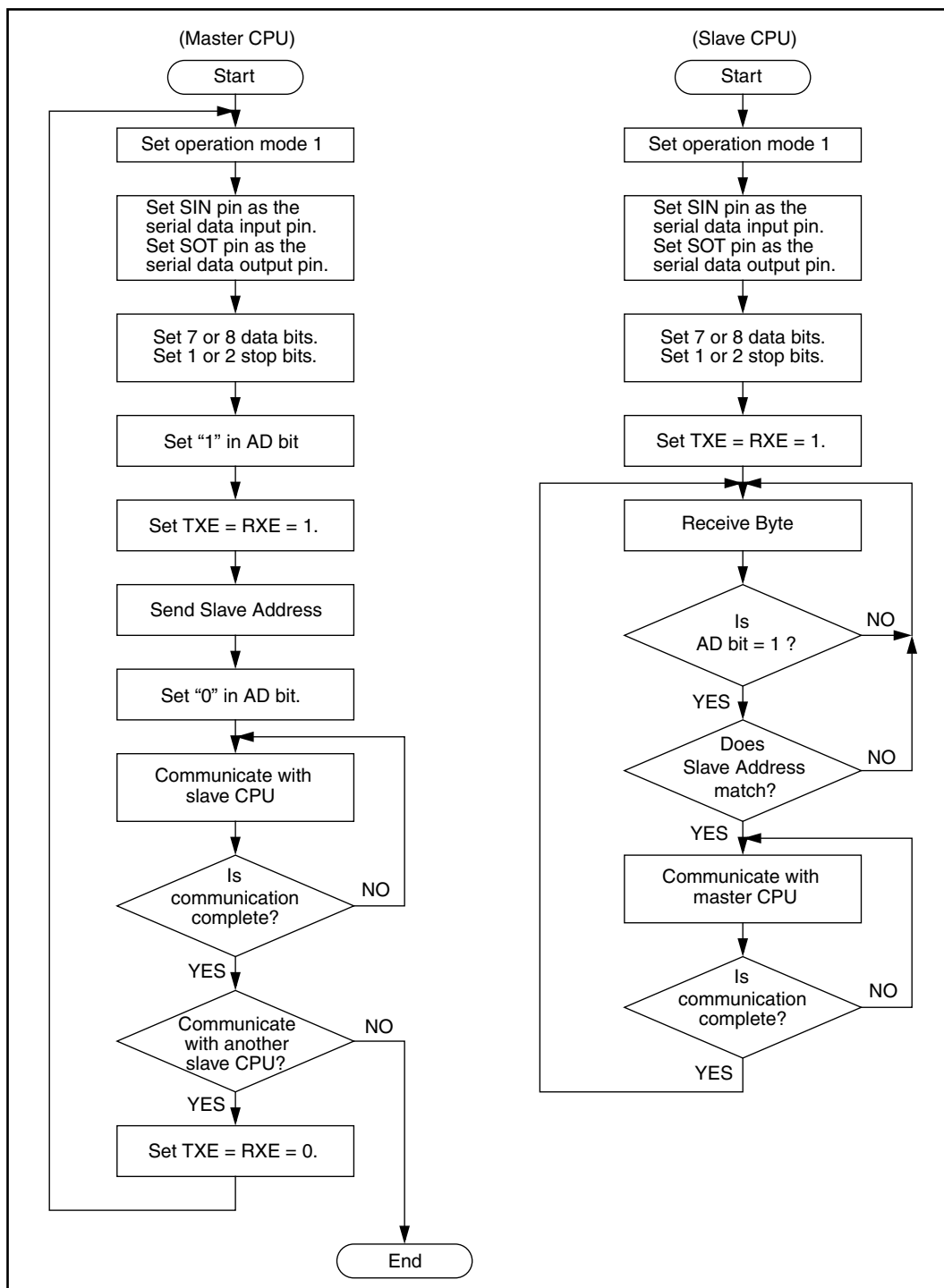
Table 32.7-3 Selection of the master-slave communication function

	Operation mode		Data	Parity	Synchroni- zation method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address trans- mission and reception	Mode 1 (send AD- bit)	Mode 1 (receive AD-bit)	AD="1" + 7- or 8-bit address	None	Asyn- chronous	1 or 2 bits	LSB or MSB first
Data trans- mission and recep- tion			AD="0" + 7- or 8-bit data				

■ Communication Procedure

When the master CPU transmits address data, communication starts. The A/D bit in the address data is set to 1, and the communication destination slave CPU is selected. Each slave CPU checks the address data using a program. When the address data indicates the address assigned to a slave CPU, the slave CPU communicates with the master CPU (ordinary data). Figure 32.7-12 shows a flowchart of master-slave communication (multiprocessor mode)

Figure 32.7-12 Master-slave communication flowchart



MB91460 Series

32.7.7 LIN Communication Function

USART communication with LIN devices is available for both LIN master or LIN slave systems.

■ LIN-Master-Slave Communication Function

The settings shown in the figure below are required to operate USART in LIN communication mode (operation mode 3).

Figure 32.7-13 Settings for USART

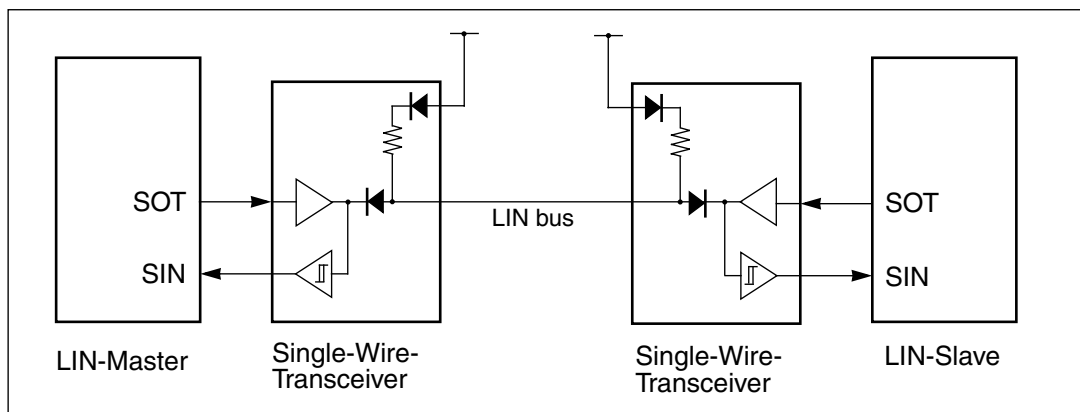
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 3	x	x	+	+	x	0	⊙	⊙	1	1	x	0	0	0	1	⊙
SSR04, TDR04/RDR04	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 3	x	⊙	⊙	⊙	⊙	+	⊙	⊙								
ESCR04,ECCR 04	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 3	⊙	⊙	⊙	⊙	x	x	x	x		⊙	x	x	x	⊙	x	x

⊙ Bit is used
 x Bit is not used
 0 / 1 Set bit to 0 / 1
 + Bit is automatically set to the correct value

■ LIN device connection

As shown in the Figure below, a communication system of one LIN-Master device and a LIN-Slave device. USART can operate both as LIN-Master or LIN-Slave.

Figure 32.7-14 Connection example of a small LIN-Bus system

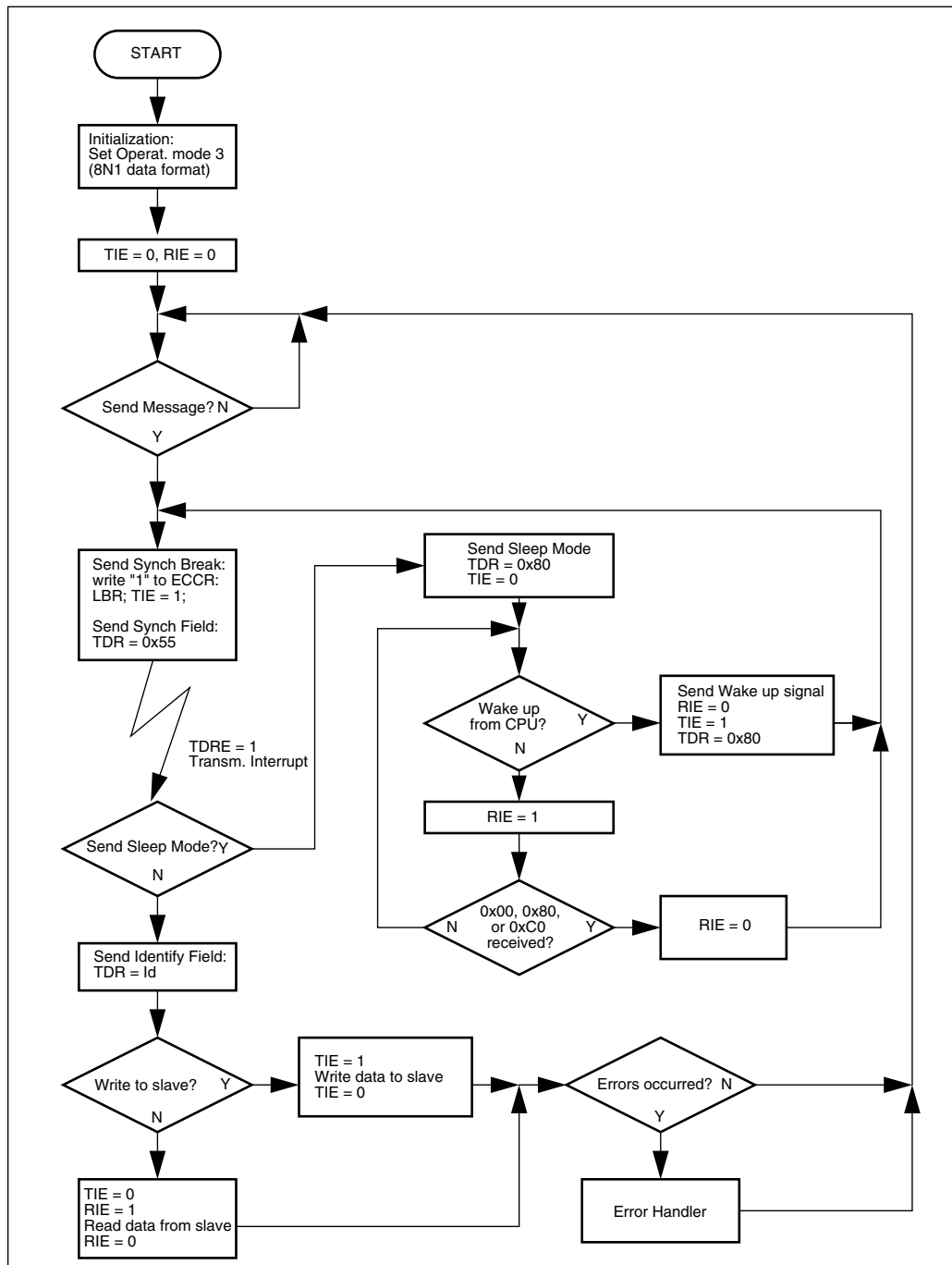


32.7.8 Sample Flowcharts for USART in LIN Communication (Operation Mode 3)

This section contains sample flowcharts for USART in LIN communication.

■ USART as master device

Figure 32.7-15 USART LIN master flow chart



■ USART as slave device

Figure 32.7-16 USART LIN slave flow chart (part1)

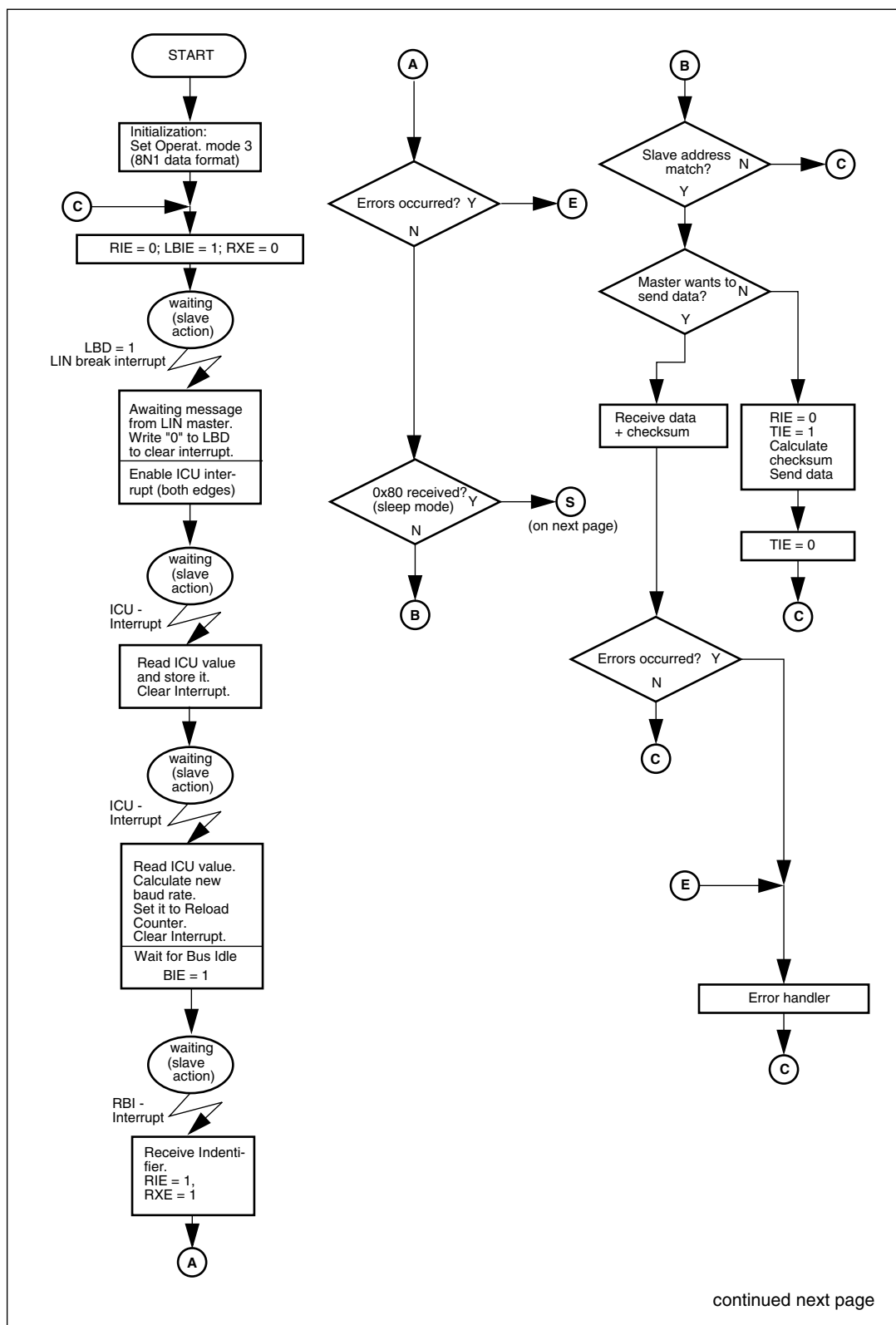
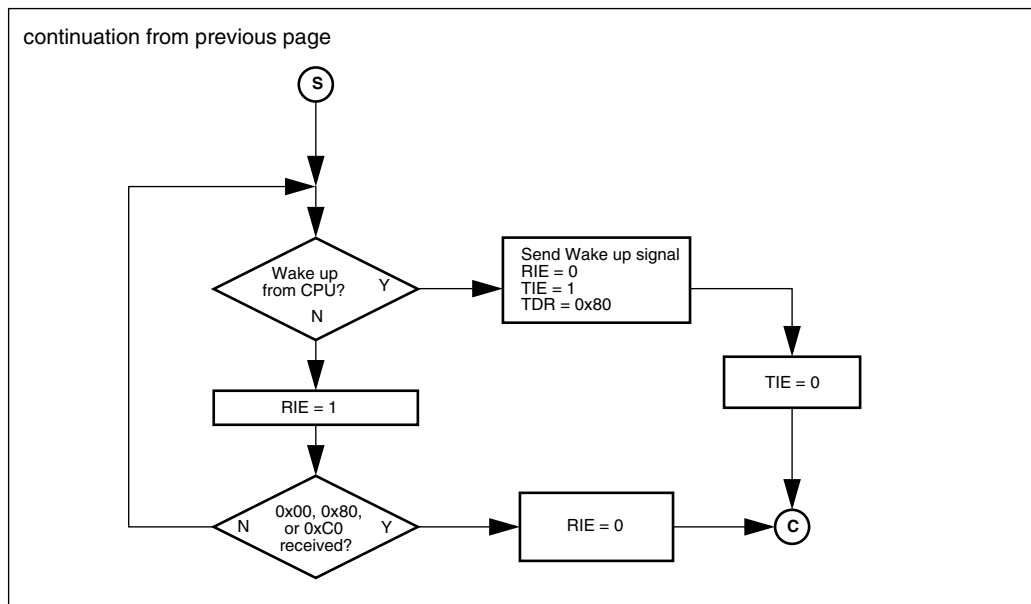


Figure 32.7-17 USART LIN slave flow chart (part 2)



32.8. Notes on using USART

Notes on using USART are given below.

■ Enabling Operations

In USART, the control register (SCR04) has TXE (transmission) and RXE (reception) operation enable bits. Both, transmission and reception operations, must be enabled before the transfer starts because they have been disabled as the default value (initial value).

In single wire bus systems like ISO 9141 (LIN bus system) because of the mono directional communication it is highly recommended to enable only one of these two bits at the same time. Because of the automatic reception the sent data by USART would be received by USART too.

■ Cancelling Transfers

Transfers can be cancelled by clearing their operation enable bits TXE / RXE. If RXE is cleared during an ongoing reception, then the USART state machines must be reset (set UPCL=1). If a transmission is ongoing at the same time, it will be cancelled too! In this case, wait until TDRE=1 and TBI=1 before setting UPCL.

■ Software reset of UART

Perform the software reset (SMR: UPCL=1), when the TXE bit of the SCR register is "0".

When performing software reset (writing '1' to SMR:UPCL) in master mode 2 (synchronous), with the mark level set to '0' (ESCR:SCES = '1'), special care has to be taken to avoid pulses on SCK.

Please stick to the following precautions:

Once: Set output data for the port function of the SCK pin to 0 by writing '0' to the related PDR register bit and enable port output function for the SCK pin by writing '1' to the related DDR register bit.

At every software reset:

Disable SCK output by writing '0' to the related PFR register bit, so that the port function outputs '0'.

Perform the software reset by writing '1' to SMR:UPCL.

Then enable SCK output again by writing '1' to the PFR register bit.

■ Clearing reception errors

Please set SCR:CRE in synchronous slave mode only, if SCR:RXE = 0.

■ Communication Mode Setting

Set the communication mode while the system is not operating. If the mode is changed during transmission or reception, the transmission or reception is stopped and possible data will be get lost.

■ Transmission Interrupt Enabling Timing

The default (initial value) of the transmission data empty flag (SSR04: TDRE) is "1" (no transmission data and transmission data write enable state). A transmission interrupt request is generated as soon as the transmission interrupt request is enabled (SSR04: TIE=1). Be sure to set the TIE bit to "1" after setting the transmission data to avoid an immediate interrupt.

■ Using LIN operation mode 3

The LIN features are also available in mode 0 (transmitting, receiving break), but using mode 3 sets the USART data format automatically to LIN format (8N1, LSB first). So, break features are applicable for bus protocols other than LIN in mode 0. Note, that the transmission time of the break is variable, but the detection is specified to a minimum of 11 serial bit times.

■ Changing Operation Settings

It is recommended to disable the communication (RXE = 0, TXE = 0), if the UART setting or mode is changed or UART is initialized.

It is strongly recommended to reset USART after changing operation settings.

Particularly in synchronous mode 2 if (for example) start/stop-bits are added to or removed from the data format.

<Caution>

If settings in the Serial Mode Register (SMR04) are desired, it is not useful to set the UPCL bit at the same time to reset USART. The correct operation settings are *not* guaranteed in this case. Thus it is recommended to set the bits of the SMR04 and *then* to set them again plus the UPCL bit.

■ LIN slave settings

To initiate USART for LIN slave make sure to set the baudrate before receiving the first LIN synchronization break. This is needed to detect safely the minimum of 11 bit times of a LIN synch break.

■ Software compatibility

Although USART is similar to older Fujitsu-UARTs it is **not** software compatible to them. The programming models may be the same, but the structure of the registers differ. Furthermore the setting of the baud rate is now determined by a reload value instead of selecting a preset value.

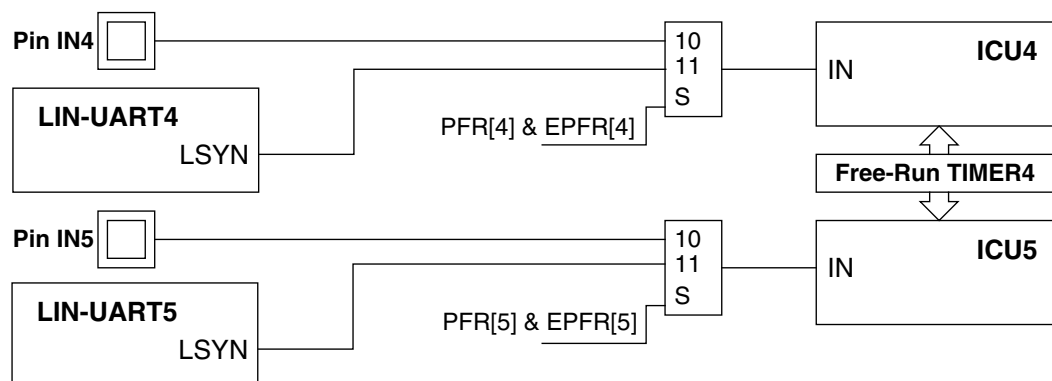
■ Bus Idle Function

The Bus Idle Function cannot be used in synchronous mode 2.

■ Baud Rate Detection Using the Input Capture Units

The USARTs provide the signal LSYN that can be connected to the ICU so that LSYN's pulse length can be measured to derive the baud rate. The connection of the LSYN signals to the ICUs is controlled by the Port 14 function register PFR and EPFR.

Figure 32.8-1 Baud Rate Detection Using the Input Capture Units



If the PFR bit equals '1' and the EPFR bits equals '0', the ICU is connected to its corresponding input pin IN.

If the PFR bit equals '1' and the EPFR bits equals '1', the USARTs are connected to the ICU.

The user has to take into account that:

- ICU4 and ICU5 share one free running timer (prescaler).

■ Effects of reception errors and CRE bit

CRE resets reception state machine and next falling edge at SINn starts reception of new byte. Therefore either set CRE bit immediately (within half bit time) after receiving errors to prevent data stream desynchronization or wait an application dependent time after receiving errors and set CRE, when SINn is idle.

Figure 32.8-2 Timing of the CRE bit

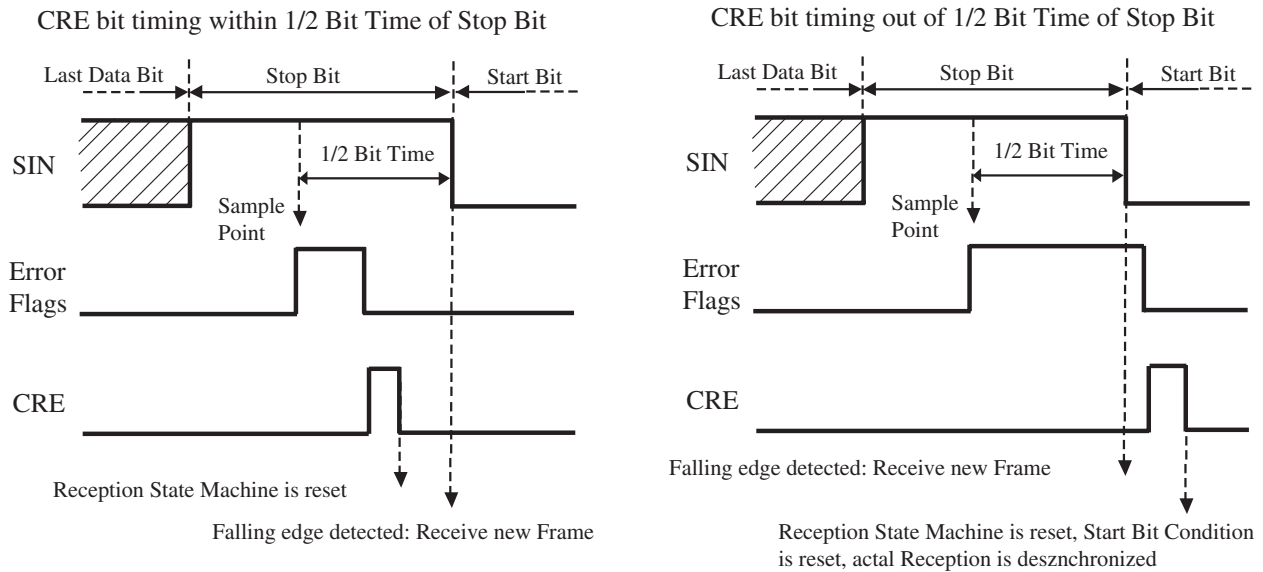
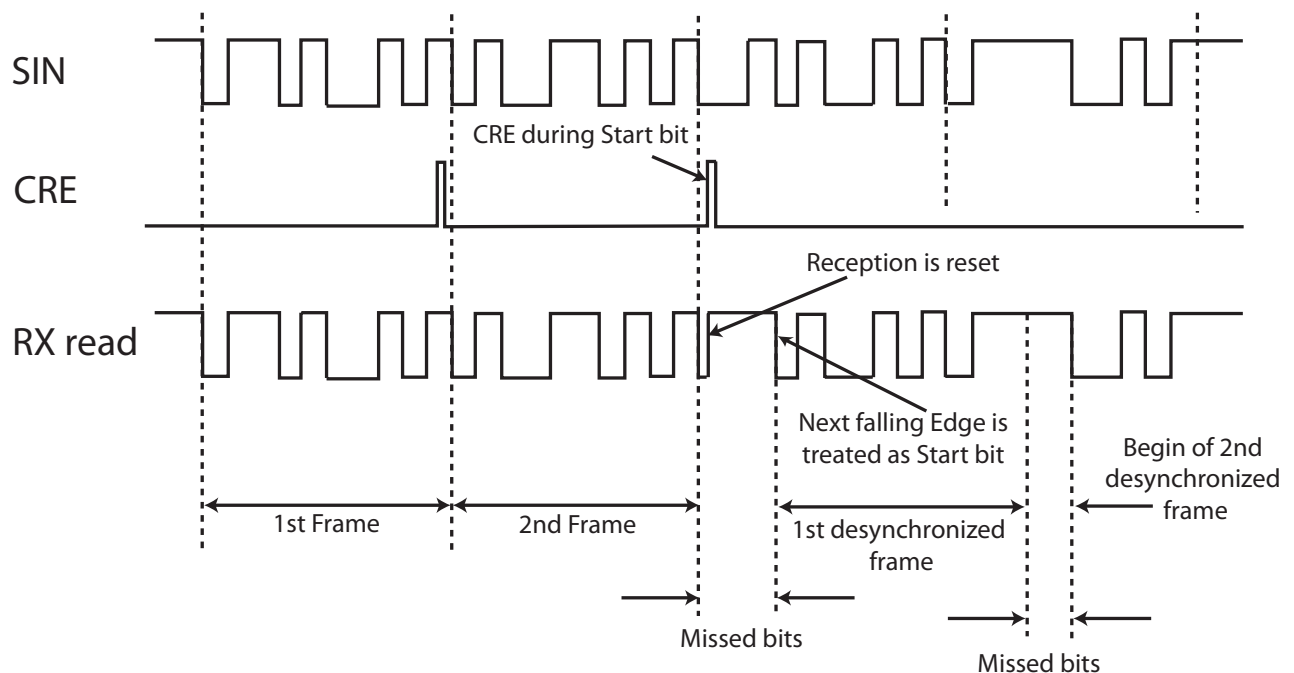


Figure 32.8-3 Data Stream Synchronization

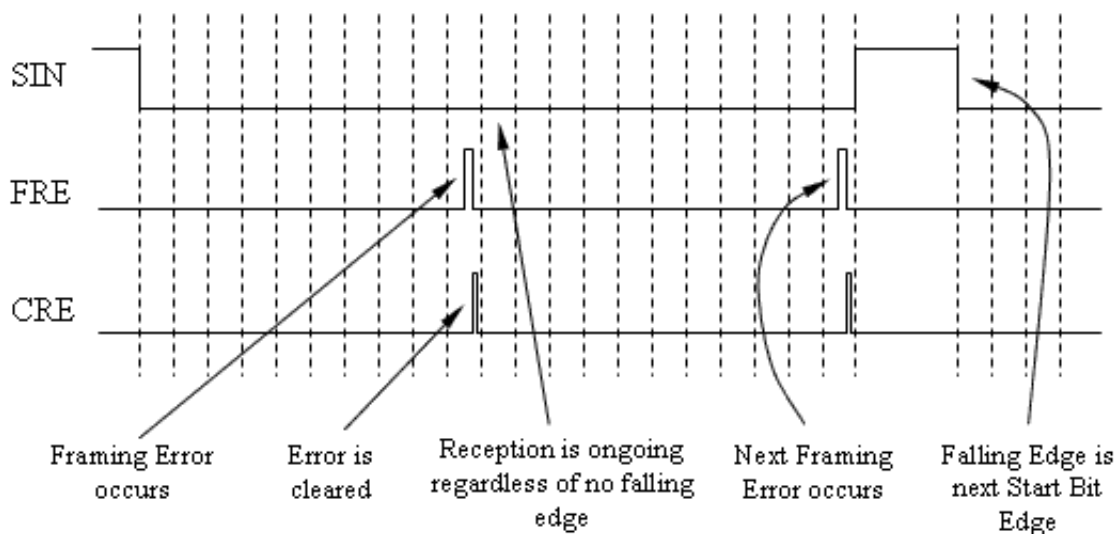
Example for Desynchronisation



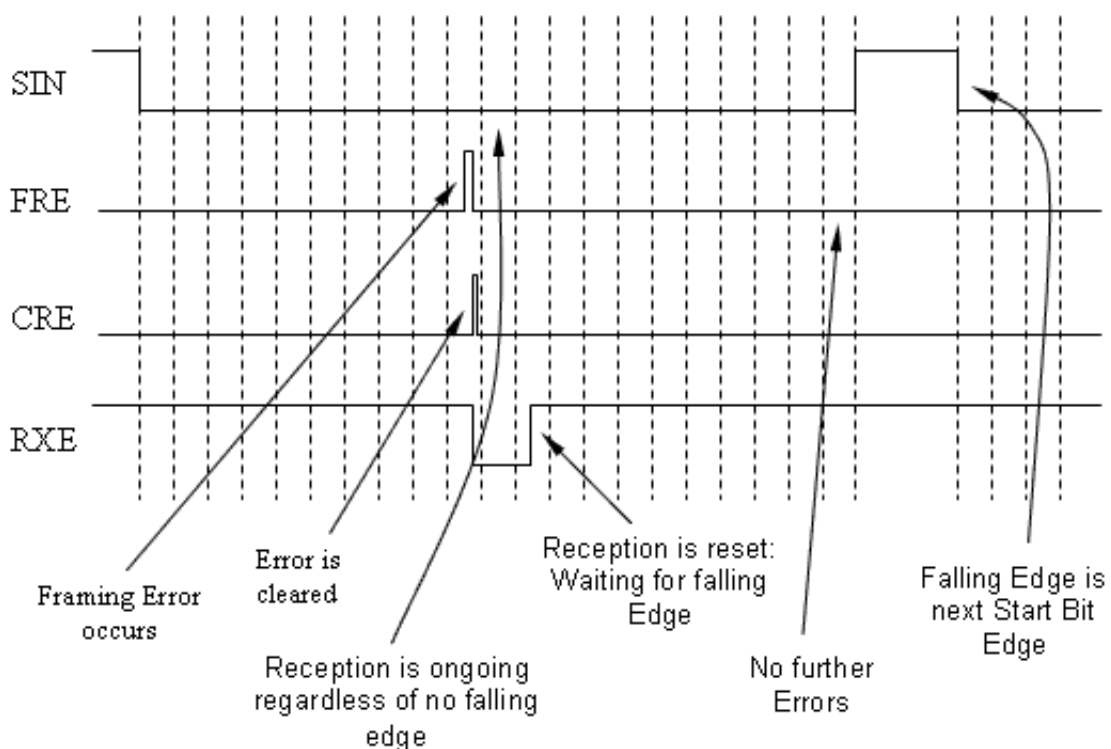
Please note, that in case a framing error occurred (stop bit: $SIN_n = "0"$) and next start bit ($SIN_n = "0"$) follows immediately, this start bit is recognized regardless of no falling edge before. This is used to remain UART synchronized to the data stream and to determine bus always dominant errors ("Fig. 32.8-4 on P. 806" upper figure) by producing next framing errors, if a recessive stop bit is expected. If this behaviour is not wanted, please disable the reception temporarily ($RXE = 1 \rightarrow 0 \rightarrow 1$) after framing error. In this case, reception goes on at next falling edge on SIN_n . ("Fig. 32.8-4 on P. 806" lower figure).

Figure 32.8-4 USART Dominant Bus Behaviour**USART Dominant Bus Behaviour**

Reception always enabled (RXE = 1)



Reception disabled temporary (RXE = 1 → 0 → 1)



Chapter 33 I²C Controller

33.1. Overview

The I2C interface is a serial I/O port supporting the Inter IC bus, operating as a master/slave device on the I2C bus.

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

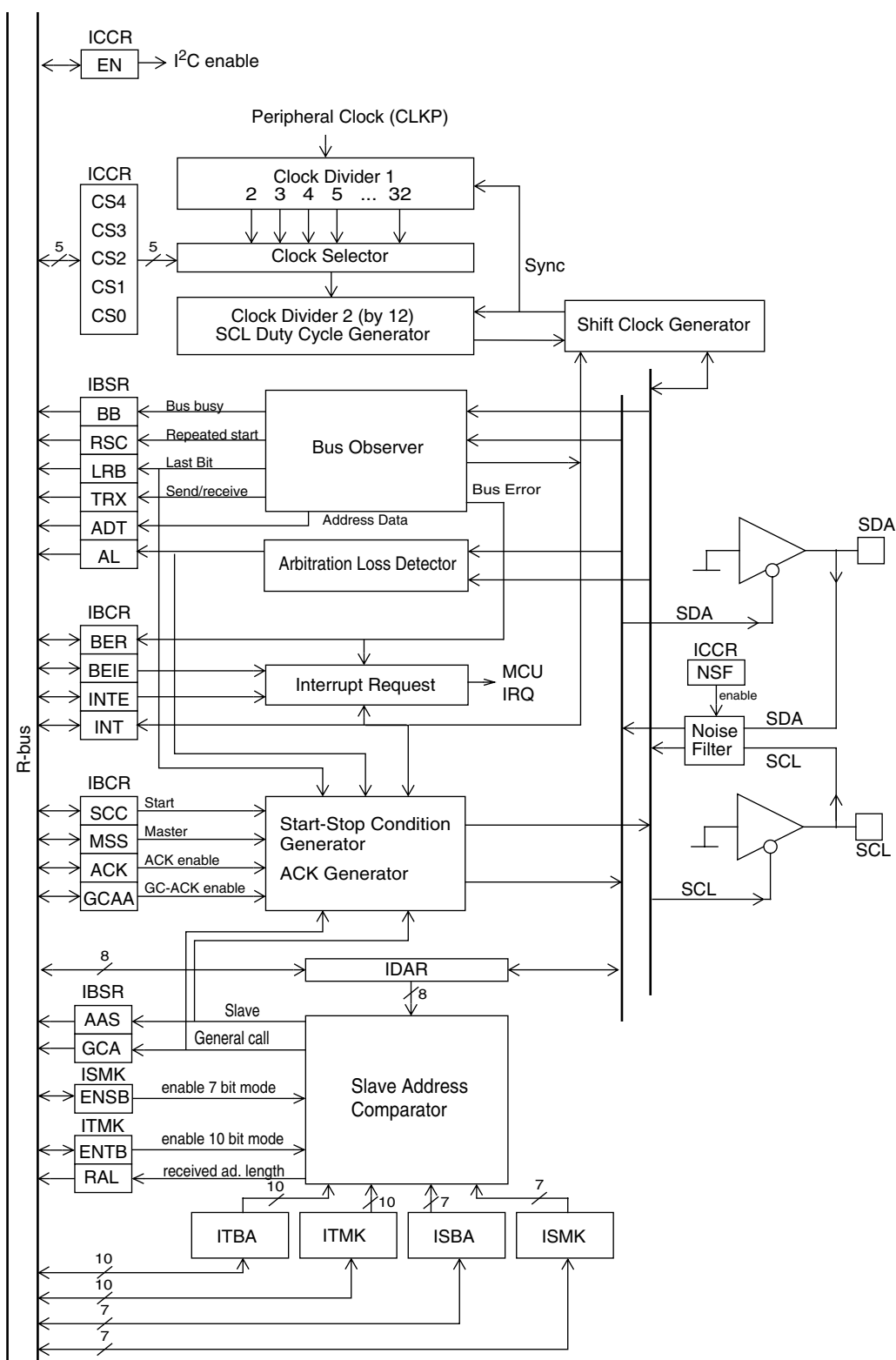
■ Features

- Master/slave transmitting and receiving functions
- Arbitration function
- Clock synchronization function
- General call addressing support
- Transfer direction detection function
- Repeated start condition generation and detection function
- Bus error detection function
- 7 bit addressing as master and slave
- 10 bit addressing as master and slave
- Possibility to give the interface a seven and a ten bit slave address
- Acknowledging upon slave address reception can be disabled (Master-only operation)
- Address masking to give interface several slave addresses (in 7 and 10 bit mode)
- Up to 400 kBit transfer rate
- Possibility to use built-in noise filters for SDA and SCL
- Reception of data at 400 kBit if peripheral clock (CLKP) is higher than 6MHz regardless of prescaler setting
- Generation MCU interrupts on transmission and bus error events
- Can be slowed down by a slave on bit and byte level

The I2C interface does not support SCL clock stretching on bit level since it can receive the full 400 kBit data rate if the peripheral clock (CLKP) is higher than 6MHz regardless of the prescaler setting. However, clock stretching on byte level is performed since SCL is pulled low during an interrupt (INT='1' in IBCR2 register).

■ Block Diagram

Figure 33.1-1 Block diagram of I2C Controller



33.2. I²C Interface Registers

This section describes the function of the I2C interface registers in detail.

Figure 33.2-1 Bus Control Register (IBCR0)

Bus control register	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D0H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR0
Read/write ⇒	(R/W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 33.2-2 Bus Status Register (IBSR0)

Bus status register	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D1H	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT	IBSR0
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 33.2-3 Ten Bit slave Address register (ITBA0)

Ten Bit Address high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D2H	---	---	---	---	---	---	TA9	TA8	ITBAH0
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit Address low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D3H	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0	ITBAL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 33.2-4 Ten bit slave address Mask register (ITMK0)

Ten Bit Address Mask high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D4H	ENTB	RAL	---	---	---	---	TM9	TM8	ITMKH0
Read/write ⇒	(R/W)	(R)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	

Ten Bit Address Mask low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D5H	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	ITMKL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Figure 33.2-5 Seven bit slave address Mask register (ISMK0)

Seven Bit Address Mask register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000D6H	ENSB	SM6	SM5	SM4	SM3	SM2	SM1	SM0	ISMK0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Figure 33.2-6 Seven Bit slave Address register (ISBA0)

Seven Bit Address register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D7H	---	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ISBA0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 33.2-7 Data Register (IDAR0)

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D9H	D7	D6	D5	D4	D3	D2	D1	D0	IDAR0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 33.2-8 Clock control register (ICCR0)

Clock Control register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000DAH	---	NSF	EN	CS4	CS3	CS2	CS1	CS0	ICCR0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	

MB91460 Series

33.2.1 Bus Control Register (IBCR0)

The bus control register (IBCR0) has the following functions:

- Interrupt request enable bits
- Interrupt generation flag
- Bus error detection flag
- Repeated start condition generation
- Master / slave mode selection
- General call acknowledge generation enabling
- Data byte acknowledge generation enabling

Write access to this register should only occur while the INT='1' or if a transfer is to be started. The user should not write to this register during an ongoing transfer since changes to the ACK or GCAA bits could result in bus errors. All bits in this register except the BER and the BEIE bit are cleared if the interface is not enabled (EN='0' in ICCR0).

Figure 33.2-9 Bus Control Register (IBCR0)

Bus control register	15	14	13	12	11	10	9	8	...	Bit no.
Address : 0000D0H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	...	IBCR0
Read/write ⇒	(R/W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	...	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	...	

[bit 15] BER (Bus Error)

This bit is the bus error interrupt flag. It is set by the hardware and cleared by the user. It is always read as '1' in a Read-Modify-Write access.

(Write access)

0	Clear bus error interrupt flag.
1	No effect.

(Read access)

0	No bus error detected.
1	One of the error conditions described below detected.

When this flag is set, the EN bit in the ICCR0 register is cleared, the I2C interface goes to pause status, data transfer is interrupted and all bits in the IBSR0 and the IBCR0 registers except BER, BEIE and INT are cleared. The BER flag must be cleared before the interface may be reenabled.

This bit is set to '1' if:

- start or stop conditions are detected at wrong places: during an address data transfer or during the transfer of the bits two to nine (acknowledge bit)
- a ten bit address header with read access is received before a ten bit write access
- a stop condition is detected while the interface is in master mode

The detection of the first two of the above conditions is enabled after the reception of the first stop condition to prevent false bus error reports if the interface is being enabled during an ongoing transfer.

[bit 14] BEIE (Bus Error Interrupt request Enable)

This bit enables the bus error interrupt request. It can only be changed by the user.

0	Bus error interrupt request disabled.
1	Bus error interrupt request enabled.

Setting this bit to '1' enables MCU interrupt request generation when the BER flag is set to '1'.

[bit 13] SCC (Start Condition Continue)

This bit is used to generate a repeated start condition. It is write only - it is always read as '0'.

0	No effect.
1	Generate repeated start condition during master transfer.

A repeated start condition is generated if a '1' is written to this bit while an interrupt in master mode (MSS='1' and INT='1') and the INT bit is cleared automatically.

[bit 12] MSS (Master Slave Select)

This is the master/slave mode selection bit. It can only be set by the user, but it can be cleared by the user and the hardware.

0	Go to slave mode.
1	Go to master mode, generate start condition and send address data byte in IDAR0 register.

It is cleared if an arbitration loss event occurs during master sending.

If a '0' is written to it during a master interrupt (MSS='1' and INT='1'), the INT bit is cleared automatically, a stop condition will be generated and the data transfer ends. Note that the MSS bit is reset immediately, the generation of the stop condition can be checked by polling the BB bit in the IBSR0 register.

If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR0 register (which should be address data) is sent.

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR2; MSS='0' in IBCR0), the interface waits until the bus is free and then starts sending.

If the interface is addressed as slave with write access (data reception) in the meantime, it will start sending after the transfer ended and the bus is free again. If the interface is sending data as slave in the meantime (AAS='1' and TRX='1' in IBSR0), it will not start sending data if the bus is free again. It is important to check whether the interface was addressed as slave (AAS='1' in IBSR0), sent the data byte successfully (MSS='1' in IBCR0) or failed to send the data byte (AL='1' in IBSR0) at the next interrupt!

[bit 11] ACK (ACKnowledge)

This is the acknowledge generation on data byte reception enable bit. It can only be changed by the user.

0	The interface will not acknowledge on data byte reception.
1	The interface will acknowledge on data byte reception.

MB91460 Series

This bit is not valid when receiving address bytes in slave mode - if the interface detects its 7 or 10 bit slave address, it will acknowledge if the corresponding enable bit (ENTB in ITMK0 or ENSB in ISMK0) is set.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in the IBSR0 register). In addition the interface must be enabled (EN='1' in ICCR0) and there has to be no bus error (BER='0' in IBCR0).

[bit 10] GCAA (General Call Address Acknowledge)

This bit enables acknowledge generation when a general call address is received. It can only be changed by the user.

0	The interface will not acknowledge on general call address byte reception.
1	The interface will acknowledge on general call address byte reception.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in IBSR0 register), write access to this bit is only possible if the interface is enabled (EN='1' in ICCR0) and if there is no bus error (BER='0' in IBCR0).

[bit 9] INTE (INTerrupt request Enable)

This bit enables the MCU interrupt request generation. It can only be changed by the user.

0	Interrupt request disabled.
1	Interrupt request enabled.

Setting this bit to '1' enables MCU interrupt request generation when the INT bit is set to '1' (by the hardware).

[bit 8]: INT (INTerrupt)

This bit is the transfer end interrupt flag. It is changed by the hardware and can be cleared by the user. It is always read as '1' in a Read-Modify-Write access.

(Write access)

0	Clear transfer end interrupt flag.
1	No effect.

(Read access)

0	Transfer not ended or not involved in current transfer or bus is idle.
1	<p>Set at the end of a 1-byte data transfer or reception including the acknowledge bit under the following conditions:</p> <ul style="list-style-type: none"> • Device is bus master. • Device is addressed as slave. • General call address received. • Arbitration loss occurred. <p>Set at the end of an address data reception (after first byte if seven bit address received, after second byte if ten bit address received) including the acknowledge bit if the device is addressed as slave.</p>

While this bit is '1' the SCL line will hold an 'L' level signal. Writing '0' to this bit clears the setting, releases the SCL line, and executes transfer of the next byte or a repeated start or stop condition is generated. Additionally, this bit is cleared if a '1' is written to the SCC bit or the MSS bit is being cleared.

SCC, MSS And INT Bit Competition

Simultaneously writing to the SCC, MSS and INT bits causes a competition to transfer the next byte, to generate a repeated start condition or to generate a stop condition. In these cases the order of priority is as follows:

Next byte transfer and stop condition generation.

When '0' is written to the INT bit and '0' is written to the MSS bit, the MSS bit takes priority and a stop condition is generated.

Next byte transfer and start condition generation.

When '0' is written to the INT bit and '1' is written to the SCC bit, the SCC bit takes priority. A repeated start condition is generated and the content of the IDAR0 register is sent.

Repeated start condition generation and stop condition generation.

When '1' is written to the SCC bit and '0' to the MSS bit, the MSS bit clearing takes priority. A stop condition is generated and the interface enters slave mode.

33.2.2 Bus Status Register (IBSR0)

The bus status register (IBSR0) has the following functions:

- Bus busy detection
- Repeated start condition detection
- Arbitration loss detection
- Acknowledge detection
- Data transfer direction indication
- Addressing as slave detection
- General call address detection
- Address data transfer detection

This register is read-only, all bits are controlled by the hardware. All bits are cleared if the interface is not enabled (EN='0' in ICCR0).

Figure 33.2-10 Bus Status Register (IBSR0)

Bus status register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D1H	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT	IBSR0
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 7] BB (Bus Busy)

This bit indicates the status of the I2C bus.

0	Stop condition detected (bus idle).
1	Start condition detected (bus in use).

This bit is set to '1' if a start condition is detected. It is reset upon a stop condition.

[bit 6] RSC (Repeated Start Condition)

This bit indicates detection of a repeated start condition.

0	Repeated start condition not detected.
1	Bus in use, repeated start condition detected.

This bit is cleared at the end of an address data transfer (ADT='0') or detection of a stop condition.

[bit 5] AL (Arbitration Loss)

This bit indicates an arbitration loss.

0	No arbitration loss detected.
1	Arbitration loss occurred during master sending.

This bit is cleared by writing '0' to the INT bit or by writing '1' to the MSS bit in the IBCR0 register.

An arbitration loss occurs if:

- the data sent does not match the data read on the SDA line at the rising SCL edge

- a repeated start condition is generated by another master in the first bit of a data byte
- the interface could not generate a start or stop condition because another slave pulled the SCL line low before

[bit 4] LRB (Last Received Bit)

This bit is used to store the acknowledge message from the receiving side at the transmitter side.

0	Receiver acknowledged.
1	Receiver did not acknowledge.

It is changed by the hardware upon reception of bit 9 (acknowledge bit) and is also cleared by a start or stop condition.

[bit 3] TRX (Transmitting data)

This bit indicates data sending operation during data transmission.

0	Not transmitting data.
1	Transmitting data.

It is set to '1':

- if a start condition was generated in master mode at the end of a first byte transfer and read access as slave or sending data as master

It is set to '0' if:

- the bus is idle (BB='0' in IBCR0)
- an arbitration loss occurred
- a '1' is written to the SCC bit during master interrupt (MSS='1' and INT='1')
- the MSS bit is cleared during master interrupt (MSS='1' and INT='1')
- the interface is in slave mode and the last transferred byte was not acknowledged
- the interface is in slave mode and it is receiving data
- the interface is in master mode and is reading data from a slave

[bit 2] AAS (Addressed As Slave)

This bit indicates detection of a slave addressing.

0	Not addressed as slave.
1	Addressed as slave.

This bit is cleared by a (repeated-) start or stop condition. It is set if the interface detects its seven and/or ten bit slave address.

[bit 1] GCA (General Call Address)

This bit indicates detection of a general call address (0x00).

0	General call address not received as slave.
---	---

1	General call address received as slave.
---	---

This bit is cleared by a (repeated-) start or stop condition.

[bit 0] ADT (Address Data Transfer)

This bit indicates the detection of an address data transfer.

0	Incoming data is not address data (or bus is not in use).
1	Incoming data is address data.

This bit is set to '1' by a start condition. It is cleared after the second byte if a ten bit slave address header with write access is detected, else it is cleared after the first byte.

"After" the first/second byte means:

- a '0' is written to the MSS bit during a master interrupt (MSS='1' and INT='1' in IBCR0)
- a '1' is written to the SCC bit during a master interrupt (MSS='1' and INT='1' in IBCR0)
- the INT bit is being cleared
- the beginning of every byte transfer if the interface is not involved in the current transfer as master or slave

33.2.3 Ten Bit Slave Address Register (ITBA0)

This register (ITBAH0 / ITBAL0) designates the ten bit slave address.

Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

Figure 33.2-11 Ten Bit Slave Address Register (ITBA0)

Ten Bit Address high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D2H	---	---	---	---	---	---	TA9	TA8	ITBAH0
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit Address low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D3H	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0	ITBAL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 15] - [bit 10] Not used.

These bits always read '0'.

[bit 9] - [bit 0] TBA - Ten Bit slave Address (TA9-TA0)

When address data is received in slave mode, it is compared to the ITBA0 register if the ten bit address is enabled (ENTB='1' in the ITMK0 register). An acknowledge is sent to the master after reception of a ten bit address header with write access¹. Then, the second incoming byte is compared to the ITBA0 register. If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.

Additionally, the interface acknowledges upon the reception of a ten bit header with read access² after a repeated start condition.

All bits of the slave address may be masked using the ITMK0 register. The received ten bit slave address is written back to the ITBA0 register, it is only valid while the AAS bit in the IBSR0 register is '1'.

1. Note: a ten bit header (write access) consists of the following bit sequence: 11110, TA9, TA8, 0.
2. Note: a ten bit header (read access) consists of the following bit sequence: 11110, TA9, TA8, 1.

33.2.4 Ten Bit Address Mask Register (ITMK0)

This register contains the ten bit slave address mask and the ten bit slave address enable bit.

Figure 33.2-12 Ten Bit Address Mask Register (ITMK0)

Ten Bit Address Mask high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D4H	ENTB	RAL	---	---	---	---	TM9	TM8	ITMKH0
Read/write ⇒	(R/W)	(R)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	

Ten Bit Address Mask low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D5H	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	ITMKL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

[bit 15] ENTB - EnaBle Ten Bit slave address

This bit enables the ten bit slave address (and the acknowledging upon its reception). Write access to this bit is only possible if the interface is disabled (EN='0' in ICCR0).

0	Ten bit slave address disabled.
1	Ten bit slave address enabled.

[bit 14] RAL - Received slave Address Length

This bit indicates whether the interface was addressed as a seven or ten bit slave. It is read-only.

0	Addressed as seven bit slave.
1	Addressed as ten bit slave.

This bit can be used to determine whether the interface was addressed as a seven or ten bit slave if both slave addresses are enabled (ENTB='1' and ENSB='1'). Its contents is only valid if the AAS bit in the IBSR0 register is '1'. This bit is also reset if the interface is disabled (EN='0' in ICCR0).

[bit 13] - [bit 10] Not used.

These bits always read '1'.

[bit 9] - [bit 0] TMK - Ten bit slave address Mask (TM9-TM0).

This register is used to mask the ten bit slave address of the interface. Write access to these bits is only possible if the interface is disabled (EN='0' in ICCR0).

0	Bit is not used in slave address comparison.
1	Bit is used in slave address comparison.

This can be used to make the interface acknowledge on multiple ten bit slave addresses. Only the bits set to '1' in this register are used in the ten bit slave address comparison. The received slave address is written back

to the ITBA0 register and thus may be determined by reading the ITBA0 register if the AAS bit in the IBSR0 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

33.2.5 Seven Bit Slave Address Register (ISBA0)

This register designates the seven bit slave address.

Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

Figure 33.2-13 Seven Bit Slave Address Register (ISBA0)

Seven Bit Address register	7	6	5	4	3	2	1	0	← Bit no.
Address : 0000D7H	---	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ISBA0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 7] Not used.

This bit always reads '0'.

[bit 6] - [bit 0] Seven Bit slave Address (SA6-SA0)

When address data is received in slave mode, it is compared to the ISBA0 register if the seven bit address is enabled (ENSB='1' in the ISMK0 register). If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.

All bits of the slave address may be masked using the ISMK0 register. The received seven bit slave address is written back to the ISBA0 register, it is only valid while the AAS bit in the IBSR0 register is '1'.

The interface does not compare the contents of this register to the incoming data if a ten bit header or a general call is received.

33.2.6 Seven Bit Slave Address Mask Register (ISMK0)

This register contains the seven bit slave address mask and the seven bit mode enable bit. Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

Figure 33.2-14 Seven Bit Slave Address Mask Register (ISMK0)

Seven Bit Address Mask register	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000D6H	ENSB	SM6	SM5	SM4	SM3	SM2	SM1	SM0	ISMK0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

[bit 15] ENSB - Enable Seven Bit slave address

This bit enables the seven bit slave address (and the acknowledge upon its reception).

0	Seven bit slave address disabled.
1	Seven bit slave address enabled.

[bit 14] - [bit 8] SMK - Seven bit slave address Mask (SM6-SM0)

This register is used to mask the seven bit slave address of the interface.

0	Bit is not used in slave address comparison.
1	Bit is used in slave address comparison.

This can be used to make the interface acknowledge on multiple seven bit slave addresses. Only the bits set to '1' in this register are used in the seven bit slave address comparison. The received slave address is written back to the ISBA0 register and thus may be determined by reading the ISBA0 register if the AAS bit in the IBSR0 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

33.2.7 Data Register (IDAR0)

Figure 33.2-15 Data Register (IDAR0)

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D9H	D7	D6	D5	D4	D3	D2	D1	D0	IDAR0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 15] - [bit 8] Not used.

These bits always read '0'.

[bit 7] - [bit 0] Data bits (D7-D0)

The data register is used in serial data transfer, and transfers data MSB-first. This register is double buffered on the write side, so that when the bus is in use (BB='1'), write data can be loaded to the register for serial transfer. The data byte is loaded into the internal transfer register if the INT bit in the IBCR0 register is being cleared or the bus is idle (BB='0' in IBSR0). In a read access, the internal register is read directly, therefore received data values in this register are only valid if INT='1' in the IBCR2 register.

33.2.8 Clock Control Register (ICCR0)

The clock control register (ICCR0) has the following functions:

- Enable IO pad noise filters
- Enable I2C interface operation
- Setting the serial clock frequency

Figure 33.2-16 Clock Control Register (ICCR0)

Clock Control register	15	14	13	12	11	10	9	8	← Bit no.
Address : 0000DAh	---	NSF	EN	CS4	CS3	CS2	CS1	CS0	ICCR0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	

[bit 15] Not used.

This bit always reads '0'.

[bit 14] IO pad NoiSe Filter enable.

This bit enables the noise filters built into the SDA and SCL IO pads.

The noise filter will suppress single spikes with a pulse width of 0 ns (minimum) and between 1 and 1.5 cycles of R-bus (maximum). The maximum depends on the phase relationship between I2C signals (SDA, SCL) and Peripheral clock.

It should be set to '1' if the interface is transmitting or receiving at datarates above 100 kBit.

[bit 13] EN (ENable)

This bit enables the I2C interface operation. It can only be set by the user but may be cleared by the user and the hardware.

0	Interface disabled.
1	Interface enabled.

When this bit is set to '0' all bits in the IBSR0 register and IBCR0 register (except the BER and BEIE bits) are cleared, the module is disabled and the I2C lines are left open. It is cleared by the hardware if a bus error occurs (BER='1' in IBCR0).

Warning: The interface immediately stops transmitting or receiving if it is being disabled. This might leave the I2C bus in an undesired state!

[bit 12] - [bit 8] CS4-0 (Clock preScaler)

These bits select the serial bitrate. They can only be changed if the interface is disabled (EN='0') or the EN bit is being cleared in the same write access.

It is determined by the following formula:

$$\text{Bitrate} = \frac{\phi}{n \cdot 12 + 18} \quad \text{Noise filter disabled} \quad n > 0; \phi : \text{Peripheral clock CLKP (set by DIVR0 register)}$$

$$\text{Bitrate} = \frac{\phi}{n \cdot 12 + 19 (+1)} \quad \text{Noise filter enabled} \quad n > 0; \phi : \text{Peripheral clock CLKP (set by DIVR0 register)} \\ (+1): \text{Inaccuracy caused by noise filter operation}$$

Note Because of the noise filter (depending on relationship between external signal and internal clock it will cause different delays) the divider in the second formula can vary between (12n + 19) and (12n + 20).

■ Prescaler settings:

Table 33.2-1 I2C Prescaler Settings

n	CS4	CS3	CS2	CS1	CS0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
...					
31	1	1	1	1	1

Do not use n=0 prescaler setting, it violates SDA/SCL timings!

The table below shows SCL frequency measurement results for the most common Peripheral clock settings and the recommended related pre-scaler settings for 100 Kbit and 400 Kbit operation.

Table 33.2-2 I2C SCL frequency versus CLKP settings

Peripheral Clock (CLKP) [MHz]	100 kBit (Noise filter disabled)		400 kBit (Noise filter enabled)	
	n	Bitrate [kBit]	n	Bitrate [kBit]
32			5	387.5
24	19	97.5	4	352.5
16	12	98	2	372
8	6	89	1	266.5

It should be noted that the measured values have been determined by examining the last 8 cycles of a transfer. This was done because the first cycle of all address or data transfers is longer than the other cycles. To be more precise: In case of an address transfer this first cycle is 3 prescaler periods longer than the other cycles, in case of a data transfer it is 4 prescaler periods longer (see figure below).

■ SCL Waveforms

Figure 33.2-17 SCL Waveforms

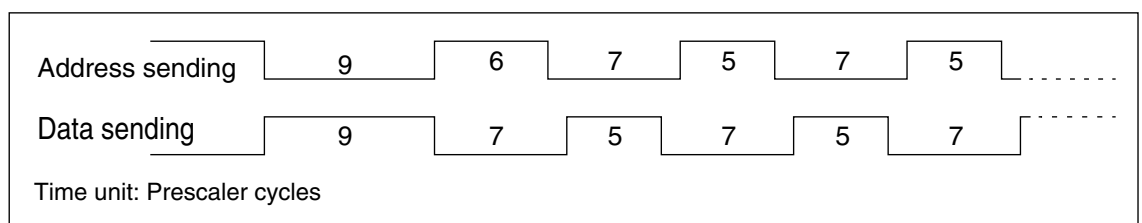


Figure 33.2-17 shows the SCL waveform for sending of address and data bits. The timings given in the figure are prescaler periods (e.g. '9' means 9 times the prescaler count based on the Peripheral clock). The timings in the figure are only valid if no other device on the I2C bus influences the SCL timing.

MB91460 Series

33.3. I²C Interface Operation

The I2C bus executes communication using two bi-directional bus lines, the serial data line (SDA) and serial clock line (SCL). The I2C interface has two open-drain I/O pins (SDA/SCL) corresponding to these lines, enabling wired logic applications.

33.3.1 Start Conditions

When the bus is free (BB='0' in IBSR0, MSS='0' in IBCR2), writing '1' to the MSS bit places the I2C interface in master mode and generates a start condition.

If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR0 register (which should be address data) is sent.

Repeated start conditions can be generated by writing '1' to the SCC bit when in bus master mode and interrupt status (MSS='1' and INT='1' in IBCR0).

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR0; MSS='0' and INT='0' in IBCR0), the interface waits until the bus is free and then starts sending.

If the interface is addressed as slave with write access (data reception) in the meantime, it will start sending after the transfer ended and the bus is free again. If the interface is sending data as slave in the meantime, it will not start sending data if the bus is free again. It is important to check whether the interface was addressed as slave (MSS='0' in IBCR0 and AAS='1' in IBSR0), sent the data byte successfully (MSS='1' in IBCR0) or failed to send the data byte (AL='1' in IBSR0) at the next interrupt!

Writing '1' to the MSS bit or SCC bit in any other situation has no significance.

33.3.2 Stop Conditions

Writing '0' to the MSS bit in master mode (MSS='1' and INT='1' in IBCR0) generates a stop condition and places the device in slave mode. Writing '0' to the MSS bit in any other situation has no significance.

After clearing the MSS bit, the interface tries to generate a stop condition which might fail if another master pulls the SCL line low before the stop condition has been generated. This will generate an interrupt after the next byte has been transferred!

33.3.3 Slave Address Detection

In slave mode, after a start condition is generated the BB is set to '1' and data sent from the master device is received into the IDAR0 register.

After the reception of eight bits, the contents of the IDAR2 register is compared to the ISBA register using the bit mask stored in ISMK0 if the ENSB bit in the ISMK0 register is '1'. If a match results, the AAS bit is set to '1' and an acknowledge signal is sent to the master. Then bit 0 of the received data (bit 0 of the IDAR0 register) is inverted and stored in the TRX bit.

If the ENTB bit in the ITMK0 register is '1' and a ten bit address header (11110, TA1, TA0, write access) is detected, the interface sends an acknowledge signal to the master and stores the inverted last data bit in the TRX register. No interrupt is generated. Then, the next transferred byte is compared (using the bit mask stored in ITMK0) to the lower byte of the ITBA0 register. If a match is found, an acknowledge signal is sent to the master, the AAS bit is set and an interrupt is generated.

If the interface was addressed as slave and detects a repeated start condition, the AAS bit is set after reception of the ten bit address header (11110, TA1, TA0, read access) and an interrupt is generated.

Since there are separate registers for the ten and seven bit address and their bitmasks, it is possible to make the interface acknowledge on both addresses by setting the ENSB (in ISMK0) and ENTB (in ITMK0) bits. The received slave address length (seven or ten bit) may be determined by reading the RAL bit in the ITMK0 register (this bit is valid if the AAS bit is set only).

It is also possible to give the interface no slave address by setting both bits to '0' if it is only used as a master. All slave address bits may be masked with their corresponding mask register (ITMK0 or ISMK0).

33.3.4 Slave Address Masking

Only the bits set to '1' in the mask registers (ITMK0 / ISMK0) are used for address comparison, all other bits are ignored. The received slave address can be read from the ITBA0 (if ten bit address received, RAL='1') or ISBA0 (if seven bit address received, RAL='0') register if the AAS bit in the IBSR0 register is '1'.

If the bitmasks are cleared, the interface can be used as a bus monitor since it will always be addressed as slave. Note that this is not a real bus monitor because it acknowledges upon any slave address reception, even if there is no other slave listening.

33.3.5 Addressing Slaves

In master mode, after a start condition is generated the BB and TRX bits are set to '1' and the contents of the IDAR0 register is sent in MSB first order. After address data is sent and an acknowledge signal was received from the slave device, bit 0 of the sent data (bit 0 of the IDAR0 register after sending) is inverted and stored in the TRX bit. Acknowledgement by the slave may be checked using the LRB bit in the IBSR0 register. This procedure also applies to a repeated start condition.

In order to address a ten bit slave for write access, two bytes have to be sent. The first one is the ten bit address header which consists of the bitsequence '1 1 1 1 0 A9 A8 0', it is followed by the second byte containing the lower eight bits of the ten bit slave address (A7 - A0).

A ten bit slave is accessed for reading by sending the above byte sequence and generating a repeated start condition (SCC bit in IBCR0) followed by a ten bit address header with read access (1 1 1 1 0 A9 A8 1).

Summary of the address data bytes:

7 bit slave, write access: Start condition - A6 A5 A4 A3 A2 A1 A0 0.

7 bit slave, read access: Start condition - A6 A5 A4 A3 A2 A1 A0 1.

10 bit slave, write access: Start condition - 1 1 1 1 0 A9 A8 0 - A7 A6 A5 A4 A3 A2 A1 A0.

10 bit slave, read access: Start condition - 1 1 1 1 0 A9 A8 1 - A7 A6 A5 A4 A3 A2 A1 A0 - repeated start - 1 1 1 1 0 A9 A8 1.

33.3.6 Arbitration

During sending in master mode, if another master device is sending data at the same time, arbitration is performed. If a device is sending the data value '1' and the data on the SDA line has an 'L' level value, the device is considered to have lost arbitration, and the AL bit is set to '1.' Also, the AL bit is set to '1' if a start condition is detected at the first bit of a data byte but the interface did not want to generate one or the generation of a start or stop condition failed by some reason.

Arbitration loss detection clears both the MSS and TRX bit and immediately places the device in slave mode so it is able to acknowledge if its own slave address is being sent.

33.3.7 Acknowledgement

Acknowledge bits are sent from the receiver to the transmitter. The ACK bit in the IBCR0 register can be used to select whether to send an acknowledgment when data bytes are received.

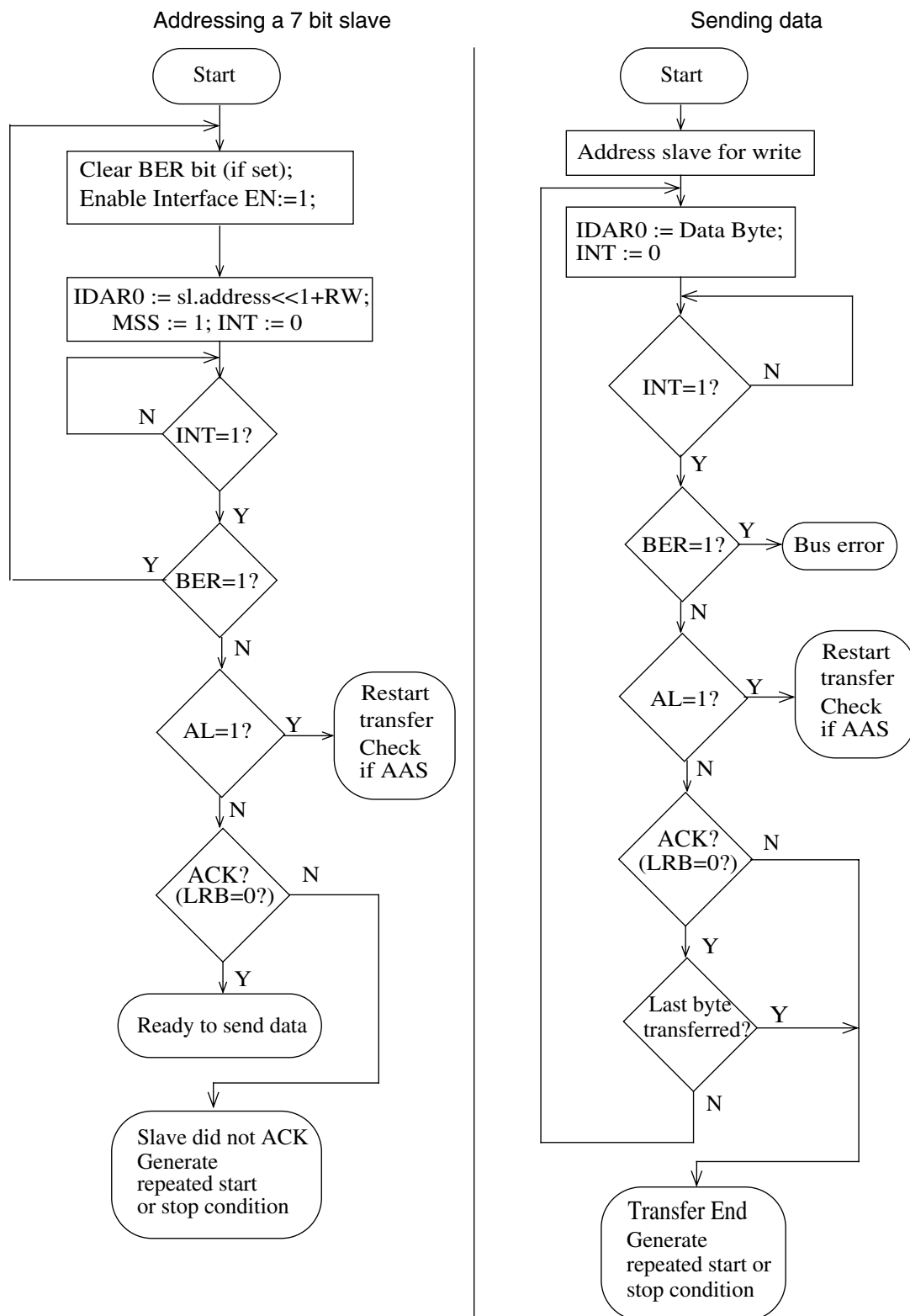
When data is send in slave mode (read access from another master), if no acknowledgement is received from the master, the TRX bit is set to '0' and the device goes to receiving mode. This enables the master to generate a stop condition as soon as the slave has released the SCL line.

In master mode, acknowledgement by the slave may be checked by reading the LRB bit in the IBSR0 register.

33.4. Programming Flow Charts

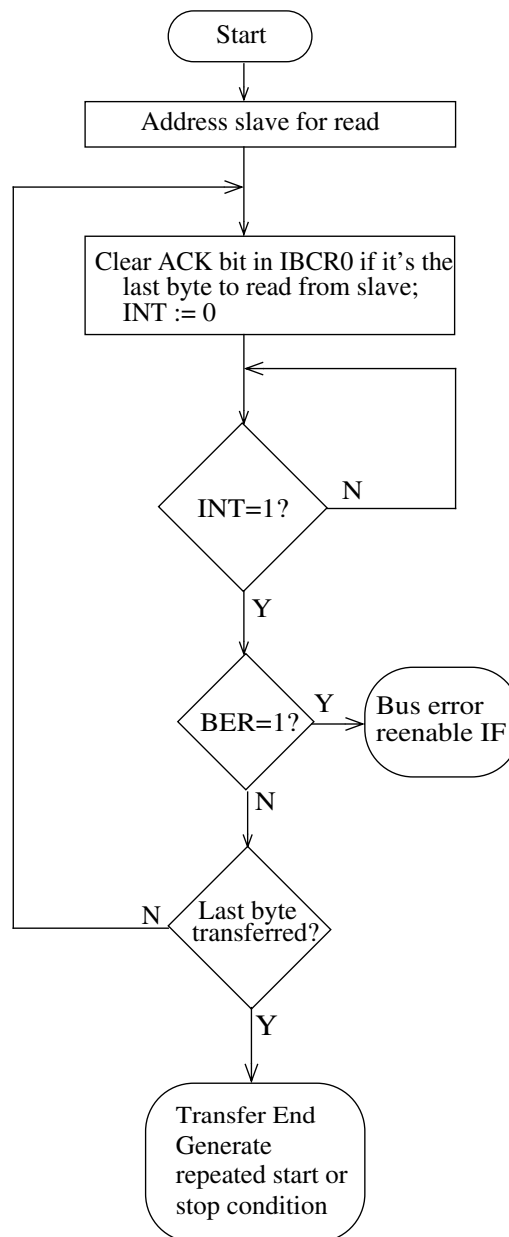
33.4.1 Example Of Slave Addressing And Sending Data

Figure 33.4-1 Example Of Slave Addressing And Sending Data



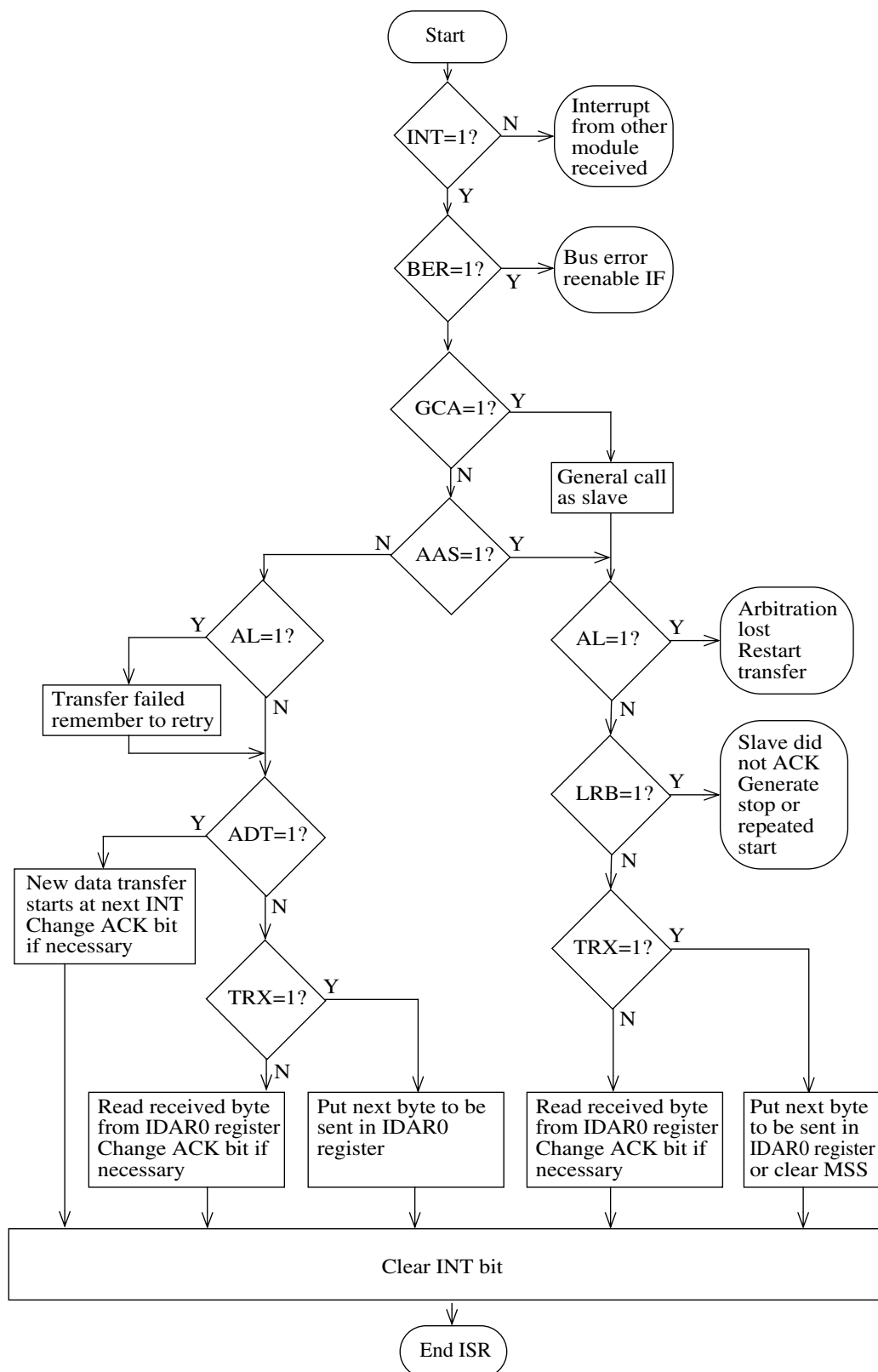
33.4.2 Example Of Receiving Data

Figure 33.4-2 Example Of Receiving Data



33.4.3 Example Of An Interrupt Handler

Figure 33.4-3 Example Of An Interrupt Handler



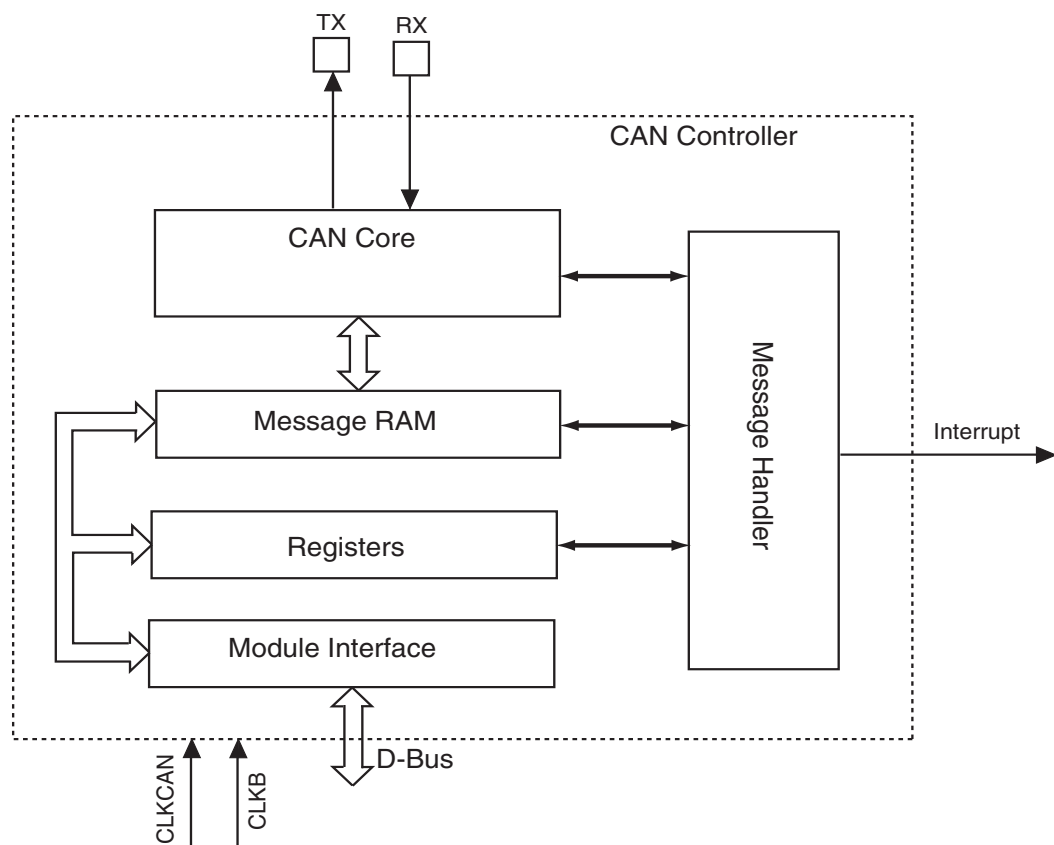
Chapter 34 CAN Controller

34.1. Overview

The CAN performs communication according to the CAN protocol version 2.0 part A and B. The bit rate can be programmed to values up to 1MBit/s. For the connection to the physical layer additional transceiver hardware is required.

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#)

Figure 34.1-1 Block diagram of the CAN Controller



■ **The CAN implements the following features:**

- Supports CAN protocol version 2.0 part A and B
- Bit rates up to 1 MBit/s
- 32 (up to 128) Message Objects
- Each Message Object has its own identifier mask
- Programmable FIFO mode (concatenation of Message Objects)
- Maskable interrupt
- Disabled Automatic Retransmission mode for Time Triggered CAN applications
- Programmable loop-back mode for self-test operation

- Operation modes: basic mode, silent mode
- Software control of TX pin is possible

■ **This chapter uses the following terms and abbreviations.**

Term	Meaning
CAN	Controller Area Network
BSP	Bit Stream Processor
BTL	Bit Timing Logic
CRC	Cyclic Redundancy Check
DLC	Data Length Code
EML	Error Management Logic
FSM	Finite State Machine
TTCAN	Time Triggered CAN

MB91460 Series

34.2. Register Description

This section lists the CAN registers and describes the function of each register in detail.

34.2.1 Programmer's Model

The CAN module allocates an address space of 256 bytes (64 words). The CAN registers can be accessed from the CPU in byte, halfword and word.

The two sets of interface registers (IF1 and IF2) control the CPU access to the Message RAM. They buffer the data to be transferred to and from the RAM, avoiding conflicts between CPU accesses and message reception/transmission.

The data registers (IF1 Data and IF2 Data) are doubled in the address map, ordered both in little endian byte and big endian byte.

If several CAN modules are present on a device then they are located linear in the address space with a constant offset of 256 bytes (64 words). The base address of each CAN module is given by the following table:

- Base-address of CAN0 : 0x00C000
- Base-address of CAN1 : 0x00C100
- Base-address of CAN2 : 0x00C200
- Base-address of CAN3 : 0x00C300
- Base-address of CAN4 : 0x00C400
- Base-address of CAN5 : 0x00C500

Table 34.2-1 CAN Register Summary

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 0x00	Control Register		Status Register		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	reserved	see descr. CTRLR	reserved	see descr. STATR	
	Reset: 0x00	Reset: 0x01	Reset: 0x00	Reset: 0x00	
Base-addr + 0x04	Error Counter		Bit Timing Register		Error Counter is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	Bit Timing Register is write enabled by CCE
	RPREC[6:0]	TEC[7:0]	TSeg2[2:0], TSeg1[3:0]	SJW[1:0], BRP[5:0]	
	Reset: 0x00	Reset: 0x00	Reset: 0x23	Reset: 0x01	
Base-addr + 0x08	Interrupt Register		Test Register		Interrupt Register is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	Test Register is write enabled by Test . r signifies the actual value of the RX pin.
	Int-Id[15:8]	Int-Id[7:0]	reserved	see descr. TESTR	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00 & 0br00000000	

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 0x0C	BRP Extension Register		Reserved		BRP Extension Register is write enabled by CCE.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	reserved	BRP[3:0]	reserved	reserved	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x10	IF1 Command Request		IF1 Command Mask		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Busy	Mess. No. [5:0]	reserved	see descr. IF1CMSK	
	Reset: 0x00	Reset: 0x01	Reset: 0x00	Reset: 0x00	
Base-addr + 0x14	IF1 Mask 2		IF1 Mask 1		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MXtd,MDir,Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset: 0xFF	Reset: 0xFF	Reset: 0xFF	Reset: 0xFF	
Base-addr + 0x18	IF1 Arbitration 2		IF1 Arbitration 1		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal,Xtd,Dir,ID[28:24]	ID[23:16]	ID[15:8]	ID[7:0]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x1C	IF1 Message Control		Reserved		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	see descr. IF1MCTR	see descr. IF1MCTR	reserved	reserved	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x20	IF1 Data A1		IF1 Data A2		Big Endian byte ordering.
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x24	IF1 Data B1		IF1 Data B2		Big Endian byte ordering.
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x30	IF1 Data A2		IF1 Data A1		Little Endian byte ordering.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	

MB91460 Series

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 0x34	IF1 Data B2		IF1 Data B1		Little Endian byte ordering.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x40	IF2 Command Request		IF2 Command Mask		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Busy	Mess. No. [5:0]	reserved	see descr. IF2CMSK	
	Reset: 0x00	Reset: 0x01	Reset: 0x00	Reset: 0x00	
Base-addr + 0x44	IF2 Mask 2		IF2 Mask 1		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MXtd,MDir,Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset: 0xFF	Reset: 0xFF	Reset: 0xFF	Reset: 0xFF	
Base-addr + 0x48	IF2 Arbitration 2		IF2 Arbitration 1		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal,Xtd,Dir,ID[28:24]	ID[23:16]	ID[15:8]	ID[7:0]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x4C	IF2 Message Control		Reserved		
	bit[15:8]	bit[7:0]	bit[7:0]	bit[15:8]	
	see descr. IF2MCTR	see descr. IF2MCTR	reserved	reserved	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x50	IF2 Data A1		IF2 Data A2		Big Endian byte ordering.
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x54	IF2 Data B1		IF2 Data B2		Big Endian byte ordering.
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x60	IF2 Data A2		IF2 Data A1		Little Endian byte ordering.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	

Address	Register				Note
	+0	+1	+2	+3	
Base-addr + 0x64	IF2 Data B2		IF2 Data B1		Little Endian byte ordering.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x80	Transmission Request Register 2		Transmission Request Register 1		Transmission Request Register is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	TxRqst[32-25]	TxRqst[24-17]	TxRqst[16-9]	TxRqst[8-1]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x84	Reserved (>32..128 Message buffer)				
Base-addr + 0x90	New Data 2		New Data 1		New Data is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	NewDat[32-25]	NewDat[24-17]	NewDat[16-9]	NewDat[8-1]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0x94	Reserved (>32..128 Message buffer)				
Base-addr + 0xA0	Interrupt Pending 2		Interrupt Pending 1		Interrupt Pending is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	IntPnd[32-25]	IntPnd[24-17]	IntPnd[16-9]	IntPnd[8-1]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0xA4	Reserved (>32..128 Message buffer)				
Base-addr + 0xB0	Message Valid 2		Message Valid 1		Message Valid is read only.
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal[32-25]	MsgVal[24-17]	MsgVal[16-9]	MsgVal[8-1]	
	Reset: 0x00	Reset: 0x00	Reset: 0x00	Reset: 0x00	
Base-addr + 0xB4	Reserved (>32..128 Message buffer)				

Figure 34.2-1 CAN Prescaler Register Summary

Address	Register				Note
	+0	+1	+2	+3	
0x04C0	CANPRE	CANCKD	-	-	CAN Prescaler
	bit[3:0]	bit[5:0]	-	-	
	CANPRE[3:0]	CANCKD[5:0]	-	-	
	Reset: 0x00	Reset: 0x00	-	-	

34.2.2 Hardware Reset Description

After hardware reset, the registers of the CAN hold the values described in [Figure 34.2-1](#).

Additionally the busoff state is reset and the output **TX** is set to recessive (HIGH). The value 0x0001 (**Init** = '1') in the CAN Control Register enables the software initialization. The CAN does not influence the CAN bus until the CPU resets **Init** to '0'.

The data stored in the Message RAM is not affected by a hardware reset. After power-on, the contents of the Message RAM is undefined.

34.2.3 CAN Protocol Related Registers

These registers are related to the CAN protocol controller in the CAN Core. They control the operating modes and the configuration of the CAN bit timing and provide status information.

■ CAN Control Register (CTRLR)

Figure 34.2-2 CAN Control Register (CTRLR)

CAN Control Register high byte								15	14	13	12	11	10	9	8	⇐ Bit no. CTRLRH
Address : Base + 0x00H								res	res	res	res	res	res	res	res	
Read/write ⇒								(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

CAN Control Register low byte								7	6	5	4	3	2	1	0	⇐ Bit no. CTRLRL
Address : Base + 0x01H								Test	CCE	DAR	res	EIE	SIE	IE	Init	
Read/write ⇒								(R/W)	(R/W)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(1)	

■ Function of the CAN Control Register (CTRLR)

[bit15 - bit8]		Reserved Bits Always write "0". Read value is not defined. Read-Modify-Write is not affected.
[bit7]	Test	Test Mode Enable
	0	Normal Operation.
	1	Test Mode.
[bit6]	CCE	Configuration Change Enable
	0	The CPU has no write access to the Bit Timing Register.
	1	The CPU has write access to the Bit Timing Register (while Init = 1)
[bit5]	DAR	Disable Automatic Retransmission
	0	Automatic Retransmission of disturbed messages enabled.
	1	Automatic Retransmission disabled.

Note: When the C_CAN is configured to work in DAR-mode (Disable Automatic Retransmission) and the host requests the transmission of several messages at the same time, only two of these messages will be transmitted. For all other requested transmit messages, the TXRQST bits will be reset, but no transmission will be started, NEWDAT and INTPND will be left unchanged. For the two messages that are transmitted, the TXRQST and NEWDAT bits will be reset and, if enabled by TXIE, INTPND will be set.

[bit4]	res	reserved bit
[bit3]	EIE	Error Interrupt request Enable
	0	Disabled - No Error Status Interrupt request will be generated.
	1	Enabled - A change in the bits BOff or EWarn in the Status Register will generate an interrupt request.
[bit2]	SIE	Status Change Interrupt request Enable
	0	Disabled - No Status Change Interrupt request will be generated.
	1	Enabled - An interrupt request will be generated when a message transfer is successfully completed or a CAN bus error is detected.
[bit1]	IE	Module Interrupt request Enable
	0	Disabled - Module Interrupt request is always inactive.
	1	Enabled - Interrupts will set the internal request. The request remains active until all pending interrupts are processed.
[bit0]	Init	Initialization
	0	Normal Operation

MB91460 Series

1 Initialization is started.

Note: The busoff recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting **Init**. If the device goes busoff, it will set **Init** of its own accord, stopping all bus activities. Once **Init** has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operations. At the end of the busoff recovery sequence, the Error Management Counters will be reset.

Note: During the waiting time after the resetting of **Init**, each time a sequence of 11 recessive bits has been monitored, a **Bit0Error** code is written to the Status Register, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the proceeding of the busoff recovery sequence.

■ Status Register (STATR)

Figure 34.2-3 Status Register (STATR)

Status Register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : Base + 0x02H	res	res	res	res	res	res	res	res	STATRH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Status Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : Base + 0x03H	BOff	EWarn	EPass	RxOK	TxOK	LEC			STATRL
Read/write ⇒	(R)	(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the Status Register (STATR)

[bit15 - bit8]		Reserved Bits
[bit7]	BOff	Busoff Status
	0	The CAN module is not busoff.
	1	The CAN module is in busoff state.
[bit6]	EWarn	Warning Status
	0	Both error counters are below the error warning limit of 96.
	1	At least one of the error counters in the EML has reached the error warning limit of 96.
[bit5]	EPass	Error Passive
	0	The CAN Core is error active.
	1	The CAN Core is in the error passive state as defined in the CAN Specification.
[bit4]	RxOk	Received a Message Successfully
	0	Since this bit was last reset by the CPU, no message has been successfully received. This bit is never reset by the CAN Core.
	1	Since this bit was last reset (to zero) by the CPU, a message has been successfully received (independent of the result of acceptance)
[bit3]	TxOk	Transmitted a Message Successfully
	0	Since this bit was reset by the CPU, no message has been successfully transmitted. This bit is never reset by the CAN Core.
	1	Since this bit was last reset by the CPU, a message has been successfully (error free and acknowledged by at least one other node) transmitted.
[bit2 - bit0]	LEC	Last Error Code (Type of the last error to occur on the CAN bus)
	0	No Error
	1	Stuff Error More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
	2	Form Error A fixed format part of a received frame has the wrong format.
	3	AckError The message this CAN Core transmitted was not acknowledged by another node.
	4	Bit1Error During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.

5	Bit0Error	During the transmission of a message (or acknowledge bit or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored Bus value was recessive. During busoff recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the busoff recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRCErr	The CRC check sum was incorrect in the message received, the CRC received for an incoming message does not match with the calculated CRC for the received data.
7	unused	When the LEC shows the value '7', no CAN bus event was detected since the CPU wrote this value to the LEC.

The **LEC** field holds a code which indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. The unused code '7' may be written by the CPU to check for updates.

■ Status Interrupts

A Status Interrupt is generated by bits **BOff** and **EWarn** (Error Interrupt) or by **RxOk**, **TxOk**, and **LEC** (Status Change Interrupt) assumed that the corresponding enable bits in the CAN Control Register are set. A change of bit **EPass** or a write to **RxOk**, **TxOk**, or **LEC** will never generate a Status Interrupt.

Reading the Status Register will clear the Status Interrupt value (8000h) in the Interrupt Register, if it is pending.

■ Error Counter (ERRCNT)

Figure 34.2-4 Error Counter (ERRCNT)

Error Counter high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : Base + 0x04H	RP	REC6-0							ERRCNTH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Error Counter low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : Base + 0x05H	TEC7-0								ERRCNTL
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the Error Counter (ERRCNT)

[bit15]	RP	Receive Error Passive
	0	The Receive Error Counter is below the error passive level.
	1	The Receive Error Counter has reached the error passive level as defined in the CAN Specification.
[bit14 - bit8]	REC6-0	Receive Error Counter
		Actual state of the Receive Error Counter. Values between 0 and 127.
[bit7 - bit0]	TEC7-0	Transmit Error Counter
		Actual state of the Transmit Error Counter. Values between 0 and 255.

■ Bit Timing Register (BTR)

Figure 34.2-5 Bit Timing Register (BTR)

Bit Timing Register high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : Base + 0x06H	res	TSeg2				TSeg1			BTRH
Read/write ⇒	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(0)	(0)	(0)	(1)	(1)	
Bit Timing Register low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : Base + 0x07H	SJW		BRP						BTRL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(1)	

MB91460 Series**■ Function of the Bit Timing Register (BTR)**

[bit15]	res	Reserved bit
[bit14 - bit12]	TSeg2	The time segment after the sample point
	0x0-0x7	Valid values for TSeg2 are [0 ... 7]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[bit11 - bit8]	TSeg1	The time segment before the sample point
	0x01-0x0F	Valid values for TSeg1 are [1 ... 15]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used
[bit7 - 6]	SJW	(Re)Synchronization Jump Width
	0x0-0x3	Valid programmed values are 0-3. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[bit5 - bit0]	BRP	Baud Rate Prescaler
	0x00-0x3F	The value by which the CLKCAN is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are[0 ... 63]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note: With a module clock **CLKCAN** of 8MHz, the reset value of 0x2301 configures the CAN for a bit rate of 500 kBit/s. The registers are only writable if bits **CCE** and **Init** in the CAN Control Register are set.

■ Test Register (TESTR)**Figure 34.2-6 Test Register (TESTR)**

Test Register high byte									← Bit no.
Address : Base + 0x0A _H									TESTRH
	15	14	13	12	11	10	9	8	
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
Test Register low byte									← Bit no.
Address : Base + 0x0B _H									TESTRL
	7	6	5	4	3	2	1	0	
Read/write ⇒	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the Test Register (TESTR)

[bit15-bit8]	res	Reserved bits
[bit7]	Rx	Monitors the actual value of the RX Pin
	0	The CAN bus is dominant (RX = '0').
	1	The CAN bus is recessive (RX = '1').
[bit6-bit5]	Tx1-0	Control of TX pin
	00	Reset value, TX is controlled by the CAN Core.
	01	Sample Point can be monitored at TX pin.
	10	TX pin drives a dominant ('0') value.
	11	TX pin drives a recessive ('1') value.
[bit4]	LBack	Loop Back Mode
	0	Loop Back Mode is disabled.
	1	Loop Back Mode is enabled.
[bit3]	Silent	Silent Mode
	0	Normal operation.
	1	The module is in Silent Mode.
[bit2]	Basic	Basic Mode
	0	Basic Mode disabled.
	1	IF1 Registers used as Tx Buffer, IF2 Registers used as Rx Buffer.
[bit1-bit0]	res	Reserved Bits

Write access to the Test Register is enabled by setting bit **Test** in the CAN Control Register. The different test functions may be combined, but **Tx1-0** unequal "00" disturbs message transfer.

MB91460 Series

■ BRP Extension Register (BRPER)

Figure 34.2-7 BRP Extension Register (BRPER)

BRP Extension Register high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : Base + 0x0Ch	res	res	res	res	res	res	res	res	BRPERH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BRP Extension Register low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : Base + 0x0Dh	res	res	res	res	BRPE				BRPERL
Read/write ⇒	(R)	(R)	(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the BRP Extension Register (BRPER)

[bit15-bit4]	res	Reserved Bits
[bit3-bit0]	BRPE	Baud Rate Prescaler Extension
0x00-0x0F		By programming BRPE the Baud Rate Prescaler can be extended to values up to 1023. The actual interpretation by the hardware is that one more than the value programmed by BRPE (MSBs) and BRP (LSBs) is used.

34.2.4 Message Interface Register Sets

There are two sets of Interface Registers which are used to control the CPU access to the Message RAM. The Interface Registers avoid conflicts between CPU access to the Message RAM and CAN message reception and transmission by buffering the data to be transferred. A complete Message Object (see [34.2.5 Message Object in the Message Memory \(Page No.852\)](#)) or parts of the Message Object may be transferred between the Message RAM and the IFx Message Buffer registers in one single transfer.

The function of the two interface register sets is identical (except for test mode **Basic**). They can be used the way that one set of registers is used for data transfer **to** the Message RAM while the other set of registers is used for the data transfer **from** the Message RAM, allowing both processes to be interrupted by each other. [Figure 34.2-8](#) gives an overview of the two Interface Register sets.

Each set of Interface Registers consists of Message Buffer Registers controlled by their own Command Registers. The Command Mask Register specifies the direction of the data transfer and which parts of a Message Object will be transferred. The Command Request Register is used to select a Message Object in the Message RAM as target or source for the transfer and to start the action specified in the Command Mask Register.

Figure 34.2-8 IF1 and IF2 Message Interface Register Sets

Address	IF1 Register Set	Address	IF2 Register Set
CAN Base + 0x10	IF1 Command Request	CAN Base + 0x40	IF2 Command Request
CAN Base + 0x12	IF1 Command Mask	CAN Base + 0x42	IF2 Command Mask
CAN Base + 0x14	IF1 Mask 2	CAN Base + 0x44	IF2 Mask 2
CAN Base + 0x16	IF1 Mask 1	CAN Base + 0x46	IF2 Mask 1
CAN Base + 0x18	IF1 Arbitration 2	CAN Base + 0x48	IF2 Arbitration 2
CAN Base + 0x1A	IF1 Arbitration 1	CAN Base + 0x4A	IF2 Arbitration 1
CAN Base + 0x1C	IF1 Message Control	CAN Base + 0x4C	IF2 Message Control
CAN Base + 0x20 CAN Base + 0x32	IF1 Data A1	CAN Base + 0x50 CAN Base + 0x62	IF2 Data A1
CAN Base + 0x22 CAN Base + 0x30	IF1 Data A2	CAN Base + 0x52 CAN Base + 0x60	IF2 Data A2
CAN Base + 0x24 CAN Base + 0x36	IF1 Data B1	CAN Base + 0x54 CAN Base + 0x66	IF2 Data B1
CAN Base + 0x26 CAN Base + 0x34	IF1 Data B2	CAN Base + 0x56 CAN Base + 0x64	IF2 Data B2

■ IFx Command Request Registers (IFxCREQ)

A message transfer is started as soon as the CPU has written the message number to the Command Request Register. With this write operation the CPU is notified that a transfer is in progress. If a CPU access to the CAN happens while the transfer is in progress then this access is delayed until the transfer has finished. After 3 to 6 **CLKCAN** periods, the transfer between the Interface Register and the Message RAM has completed and the upcoming CPU access is executed.

Figure 34.2-9 IFx Command Request Registers (IFxCREQ)

IFx Command Request Register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : Base + 0x10H & Base + 0x40H	BUSY	res	res	res	res	res	res	res	IFxCREQH
Read/write ⇒	(R/(W))	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

IFx Command Request Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : Base + 0x11H & Base + 0x41H	Message Number								IFxCREQH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(1)	

■ Function of the IFx Command Request Registers (IFxCREQ)

[bit15]	BUSY	Busy Flag
	0	Reset to zero when read/write action has finished
	1	Set to one when writing to the IFx Command Request Register
[bit14-bit8]	res	Reserved Bits
[bit5-bit0]		Message Number (for 32 message buffer CANs)
	0x00	Not a valid Message Number, interpreted as 0x20.
	0x01-0x20	Valid Message Number, the Message Object in the Message RAM is selected for data transfer.
	0x21-0x3F	Not a valid Message Number, interpreted as 0x01-0x1F.
[bit7-bit0]		Message Number (for 128 message buffer CANs)
	0x00	Not a valid Message Number, interpreted as 0x80.
	0x01-0x80	Valid Message Number, the Message Object in the Message RAM is selected for data transfer.
	0x81-0xFF	Not a valid Message Number, interpreted as 0x01-0x7F.

Note: The Busy Flag can only be used in BASIC mode (see [34.3.8 Basic Mode \(Page No.864\)](#)). When using the Message RAM (BASIC=0) the hardware interface controls the access for read and write and this flag is always read as '0'.

Note: When a **Message Number** that is not valid is written into the Command Request Register, the **Message Number** will be transformed into a valid value and that Message Object will be transferred.

■ IFx Command Mask Register (IFxCMSK)

The control bits of the IFx Command Mask Register specify the transfer direction and select which of the IFx Message Buffer Registers are source or target of the data transfer.

Figure 34.2-10 IFx Command Mask Register (IFxCMSK)

IFx Command Mask Register high byte									← Bit no.
Address : Base + 0x12H & Base + 0x42H	15	14	13	12	11	10	9	8	
	res	res	res	res	res	res	res	res	IFxCMSKH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
IFx Command Mask Register low byte									← Bit no.
Address : Base + 0x13H & Base + 0x43H	7	6	5	4	3	2	1	0	
	WR/RD	Mask	Arb	Control	CIP	TxReq/ NewDat	Data A	Data B	IFxCMSKL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the IFx Command Mask Register (IFxCMSK)

[bit15-bit8]	res	Reserved Bits
[bit7]	WR/RD	Write / Read
	0	Read: Transfer data from the Message Object addressed by the Command Request Register into the selected Message Buffer Registers.
	1	Write: Transfer data from the selected Message Buffer Registers to the Message Object addressed by the Command Request Register.

The other bits of IFx Command Mask Register have different functions depending on the transfer direction :

• Direction = Write

[bit6]	Mask	Access Mask Bits
	0	Mask bits unchanged.
	1	Transfer Identifier Mask + MDir + MXtd to Message Object.
[bit5]	Arb	Access Arbitration Bits
	0	Arbitration bits unchanged.
	1	Transfer Identifier + Dir + Xtd + MsgVal to Message Object.
[bit4]	Control	Access Control Bits
	0	Control Bits unchanged.
	1	Transfer Control Bits to Message Object.
[bit3]	CIP	Clear Interrupt Pending Bit
		When writing to a Message Object, this bit is ignored.
[bit2]	TxReq/ NewDat	Access Transmission Request Bit
	0	TxRqst bit unchanged
	1	set TxRqst bit
[bit1]	Data A	Access Data Bytes 0-3
	0	Data Bytes 0-3 unchanged.
	1	Transfer Data Bytes 0-3 to Message Object.
[bit0]	Data B	Access Data Bytes 4-7

MB91460 Series

0	Data Bytes 4-7 unchanged.
1	Transfer Data Bytes 4-7 to Message Object.

Note: If a transmission is requested by programming bit **TxRqst/NewDat** in the IFx Command Mask Register, bit **TxRqst** in the IFx Message Control Register will be ignored.

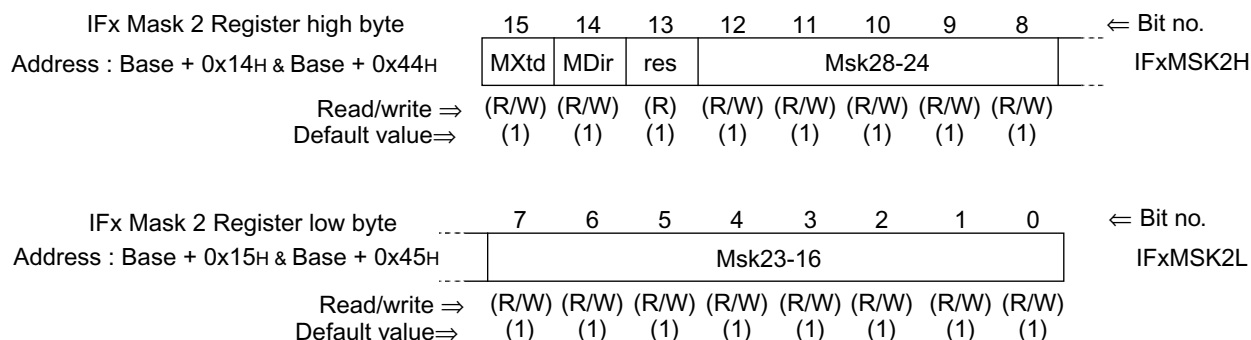
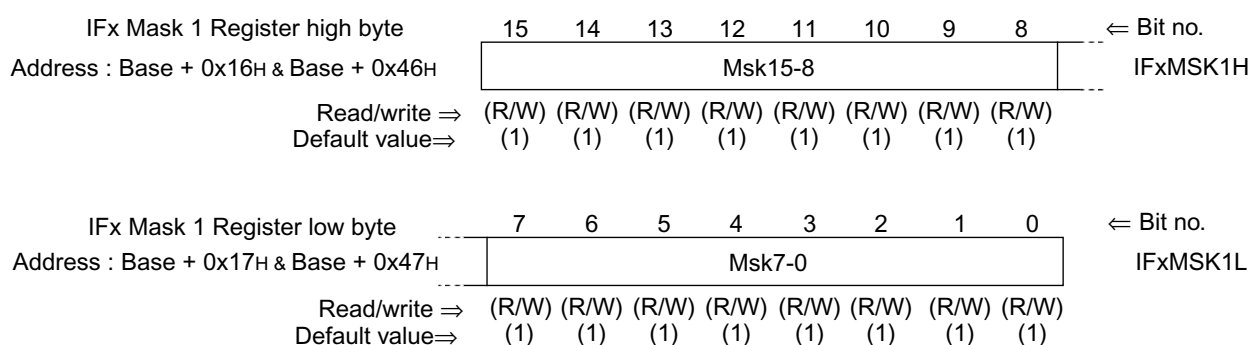
- **Direction = Read**

[bit6]	Mask	Access Mask Bits
	0	Mask bits unchanged.
	1	Transfer Identifier Mask + MDir + MXtd to IFx Message Buffer Register.
[bit5]	Arb	Access Arbitration Bits
	0	Arbitration bits unchanged.
	1	Transfer Identifier + Dir + Xtd + MsgVal to IFx Message Buffer Register.
[bit4]	Control	Access Control Bits
	0	Control Bits unchanged.
	1	Transfer Control Bits to IFx Message Buffer Register.
[bit3]	CIP	Clear Interrupt Pending Bit
	0	IntPnd bit remains unchanged.
	1	Clear IntPnd bit in the Message Object.
[bit2]	TxReq/ NewDat	Access New Data Bit
	0	NewDat bit remains unchanged.
	1	Clear NewDat bit in the Message Object.
[bit1]	Data A	Access Data Bytes 0-3
	0	Data Bytes 0-3 unchanged.
	1	Transfer Data Bytes 0-3 to IFx Message Buffer Register.
[bit0]	Data B	Access Data Bytes 4-7
	0	Data Bytes 4-7 unchanged.
	1	Transfer Data Bytes 4-7 to IFx Message Buffer Register.

Note: A read access to a Message Object can be combined with the reset of the control bits **IntPnd** and **NewDat**. The values of these bits transferred to the IFx Message Control Register always reflect the status before resetting these bits.

■ IFx Mask Registers (IFxMSK)

The bits of the Message Buffer registers mirror the Message Objects in the Message RAM.

Figure 34.2-11 IFx Mask 2 Register (IFxMSK2)**Figure 34.2-12 IFx Mask 1 Register (IFxMSK1)**

■ IFx Arbitration Registers (IFxARB)

The bits of the Message Buffer registers mirror the Message Objects in the Message RAM.

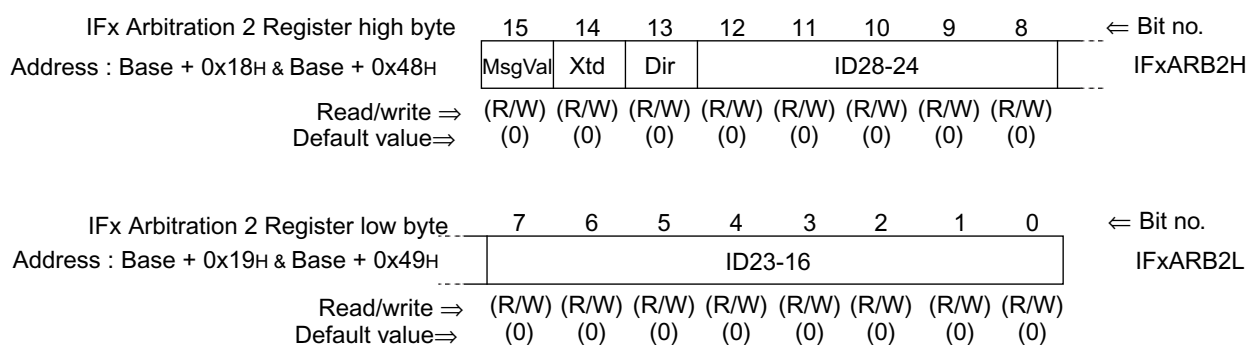
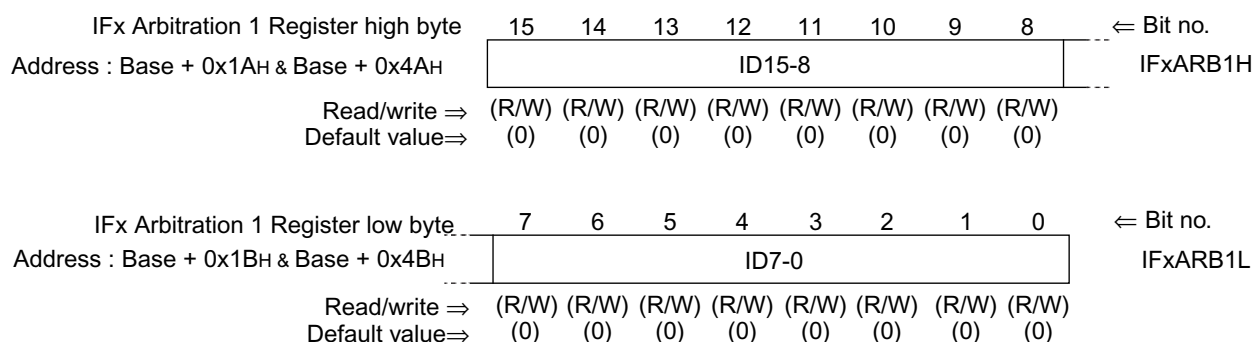
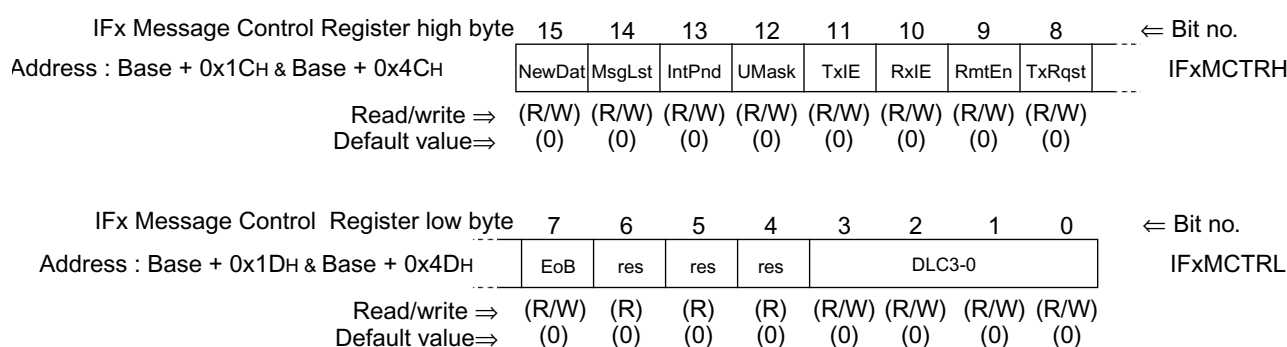
Figure 34.2-13 IFx Arbitration 2 Register (IFxARB2)

Figure 34.2-14 IFx Arbitration 1 Register (IFxARB1)



IFx Message Control Register (IFxMCTR)

Figure 34.2-15 IFx Message Control Register (IFxMCTR)



IFx Data A and Data B Registers (IFxDTA, IFxDTB)

The data bytes of CAN messages are stored in the IFx Message Buffer Registers in the following order:

	addr+0	addr+1	addr+2	addr+3
IFx Message Data A1 (addresses 0x20 & 0x50)	Data(0)	Data(1)		
IFx Message Data A2 (addresses 0x22 & 0x52)			Data(2)	Data(3)
IFx Message Data B1 (addresses 0x24 & 0x54)	Data(4)	Data(5)		
IFx Message Data B2 (addresses 0x26 & 0x56)			Data(6)	Data(7)
IFx Message Data A2 (addresses 0x30 & 0x60)	Data(3)	Data(2)		
IFx Message Data A1 (addresses 0x32 & 0x62)			Data(1)	Data(0)
IFx Message Data B2 (addresses 0x34 & 0x64)	Data(7)	Data(6)		
IFx Message Data B1 (addresses 0x36 & 0x66)			Data(5)	Data(4)

In a CAN Data Frame, Data(0) is the first, Data(7) is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

34.2.5 Message Object in the Message Memory

There are 32 Message Objects (up to 128 depending on the implementation) in the Message RAM. To avoid conflicts between CPU access to the Message RAM and CAN message reception and transmission, the CPU cannot directly access the Message Objects, these accesses are handled via the IFx Interface Registers. Figure 34.2-16 gives an overview of the two structures of a Message Object.

Figure 34.2-16 Structure of a Message Object in the Message Memory

Message Object												
UMask	Msk28-0	MXtd	MDir	EoB	NewDat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28-0	Xtd	Dir	DLC3-0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

MsgVal Message Valid

- 0 The Message Object is ignored by the Message Handler.
- 1 The Message Object is configured and should be considered by the Message Handler.

Note: The CPU must reset the **MsgVal** bit of all unused Message Objects during the initialization before it resets bit **Init** in the CAN Control Register. This bit must also be reset before the identifier **ID28-0**, the control bits **Xtd**, **Dir**, or the Data Length Code **DLC3-0** are modified, or if the Message Object is no longer required.

UMask Use Acceptance Mask

- 0 Mask ignored.
- 1 Use Mask (Msk28-0, MXtd, and MDir) for acceptance filtering.

Note: If the **UMask** bit is set to one, the Message Object's mask bits have to be programmed during initialization of the Message Object before **MsgVal** is set to one.

ID28-0 Message Identifier

- ID28 - ID0 29-bit Identifier ("Extended Frame").
- ID28 - ID18 11-bit Identifier ("Standard Frame").

Msk28-0 Identifier Mask

- 0 The corresponding bit in the identifier of the message object cannot inhibit the match in the acceptance filtering.
- 1 The corresponding identifier bit is used for acceptance filtering.

Xtd Extended Identifier

- 0 The 11-bit ("standard") Identifier will be used for this Message Object.
- 1 The 29-bit ("extended") Identifier will be used for this Message Object.

MB91460 Series**MXtd** Mask Extended Identifier

- 0 The extended identifier bit (IDE) has no effect on the acceptance filtering
- 1 The extended identifier bit (IDE) is used for acceptance filtering.

Note: When 11-bit (“standard”) Identifiers are used for a Message Object, the identifiers of received Data Frames are written into bits **ID28** to **ID18**. For acceptance filtering, only these bits together with mask bits **Msk28** to **Msk18** are considered.

Dir Message Direction

- 0 Direction = receive: On TxRqst, a Remote Frame with the identifier of this Message Object is transmitted. On reception of a Data Frame with matching identifier, that message is stored in this Message Object.
- 1 Direction = transmit: On TxRqst, the respective Message Object is transmitted as a Data Frame. On reception of a Remote Frame with matching identifier, the TxRqst bit of this Message Object is set (if RmtEn = 1).

MDir Mask Message Direction

- 0 The message direction bit (Dir) has no effect on the acceptance filtering.
- 1 The message direction bit (Dir) is used for acceptance filtering.

The Arbitration Registers **ID28-0**, **Xtd**, and **Dir** are used to define the identifier and type of outgoing messages and are used (together with the mask registers **Msk28-0**, **MXtd**, and **MDir**) for acceptance filtering of incoming messages. A received message is stored into the valid Message Object with matching identifier and Direction= receive (Data Frame) or Direction= transmit (Remote Frame). Extended frames can be stored only in Message Objects with **Xtd** = 1, standard frames in Message Objects with **Xtd** = 0. If a received message (Data Frame or Remote Frame) matches with more than one valid Message Object, it is stored into that with the lowest message number. For details see [34.4.5 Acceptance Filtering of Received Messages \(Page No.867\)](#).

EoB End of Buffer

- 0 Message Object belongs to a FIFO Buffer and is not the last Message Object of that FIFO Buffer.
- 1 Single Message Object or last Message Object of a FIFO Buffer.

Note: This bit is used to concatenate two ore more Message Objects (up to 32) to build a FIFO Buffer. **For single Message Objects (not belonging to a FIFO Buffer) this bit must always be set to 1.** For details on the concatenation of Message Objects see [34.4.13 Configuration of a FIFO Buffer \(Page No.870\)](#)

NewDat New Data

- 0 No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the CPU.
- 1 The Message Handler or the CPU has written new data into the data portion of this Message Object.

MsgLst Message Lost (only valid for Message Objects with direction = receive)

- 0 No message lost since last time this bit was reset by the CPU.
- 1 The Message Handler stored a new message into this object when NewDat was still set, the CPU has lost a message.

RxIE Receive Interrupt request Enable

- 0 IntPnd will be left unchanged after a successful reception of a frame.
- 1 IntPnd will be set after a successful reception of a frame.

TxIE Transmit Interrupt request Enable

- 0 IntPnd will be left unchanged after the successful transmission of a frame.
- 1 IntPnd will be set after a successful transmission of a frame.

IntPnd Interrupt Pending

- 0 This message object is not the source of an interrupt.
- 1 This message object is the source of an interrupt. The Interrupt Identifier in the Interrupt Register will point to this message object if there is no other interrupt source with higher priority.

RmtEn Remote Enable

- 0 At the reception of a Remote Frame, TxRqst is left unchanged.
- 1 At the reception of a Remote Frame, TxRqst is set.

TxRqst Transmit Request

- 0 This Message Object is not waiting for transmission.
- 1 The transmission of this Message Object is requested and is not yet done.

Note: When the lowest priority message buffer is used for transmission, setting **TxRqst** = "0" may cause a delay of transmission, when **TxRqst** is set to "1" again.

Depending on the exact time when **TxRqst** was set to 0, the message may not be transmitted immediately after setting **TxRqst** = "1", but after any of the following events:

1. there is activity ongoing on the CANbus
2. a transmission request is issued on another message object
3. the CANbus is initialized by the INIT bit

In general, there is no need to cancel an ongoing transmission by setting TxRqst = 0. If the content of a message object needs to be changed while TxRqst = 1, it is sufficient to update the message object via the CPU interface registers (Identifier, **DLC**, **Data**, with **TxRqst** and **NewDat**, optionally **TxIE**). The updated content will be transmitted at the next opportunity.

DLC3-0 Data Length Code

- 0-8 Data Frame has 0-8 data bytes.
- 9-15 Data Frame has 8 data bytes.

Note: The Data Length Code of a Message Object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the Message Handler stores a data frame, it will write the

MB91460 Series

DLC to the value given by the received message.

Data 0	1st data byte of a CAN Data Frame
Data 1	2nd data byte of a CAN Data Frame
Data 2	3rd data byte of a CAN Data Frame
Data 3	4th data byte of a CAN Data Frame
Data 4	5th data byte of a CAN Data Frame
Data 5	6th data byte of a CAN Data Frame
Data 6	7th data byte of a CAN Data Frame
Data 7	8th data byte of a CAN Data Frame

Note: Byte **Data 0** is the first data byte shifted into the shift register of the CAN Core during a reception, byte **Data 7** is the last. When the Message Handler stores a Data Frame, it will write all the eight data bytes into a Message Object. If the Data Length Code is less than 8, the remaining bytes of the Message Object will be overwritten by **non specified values**.

34.2.6 Message Handler Registers

All Message Handler registers are read-only. Their contents (**TxRqst**, **NewDat**, **IntPnd**, and **MsgVal** bits of each Message Object and the Interrupt Identifier) is status information provided by the Message Handler FSM.

■ Interrupt Register (INTR)

Figure 34.2-17 Interrupt Register (INTR)

Interrupt Register high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : Base + 0x08H	IntId15-8								INTRH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
Interrupt Register low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : Base + 0x09H	IntId7-0								INTRL
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Function of the Interrupt Register (INTR)

- (For 32 message buffer CANs)

IntId15-0 Interrupt Identifier (the number here indicates the source of the interrupt)

0x0000	No interrupt is pending.
0x0001-0x0020	Number of Message Object which caused the interrupt.
0x0021-0x7FFF	unused.
0x8000	Status Interrupt.

0x8001- unused.
0xFFFF

- (For 128 message buffer CANs)

IntId15-0 Interrupt Identifier (the number here indicates the source of the interrupt)

0x0000 No interrupt is pending.
0x0001- Number of Message Object which caused the interrupt.
0x0080
0x0081- unused.
0x7FFF
0x8000 Status Interrupt.
0x8001- unused.
0xFFFF

If several interrupts are pending, the CAN Interrupt Register will point to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the CPU has cleared it. If **IntId** is different from 0x0000 and **IE** is set, the interrupt line to the CPU is active. The interrupt line remains active until **IntId** is back to value 0x0000 (the cause of the interrupt is reset) or until **IE** is reset.

The Status Interrupt has the highest priority. Among the message interrupts, the Message Object's interrupt priority decreases with increasing message number.

A message interrupt is cleared by clearing the Message Object's **IntPnd** bit. The Status Interrupt is cleared by reading the Status Register.

■ Transmission Request Registers (TREQR)

Figure 34.2-18 Transmission Request Register 2 (TREQR2)

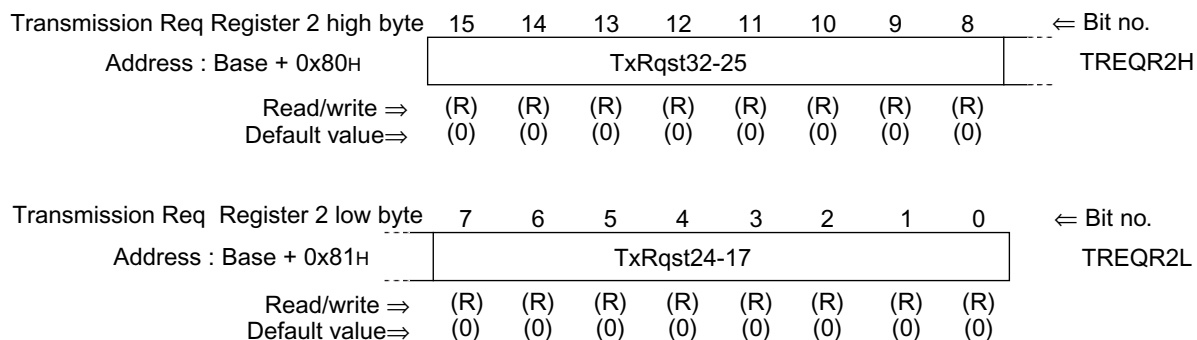
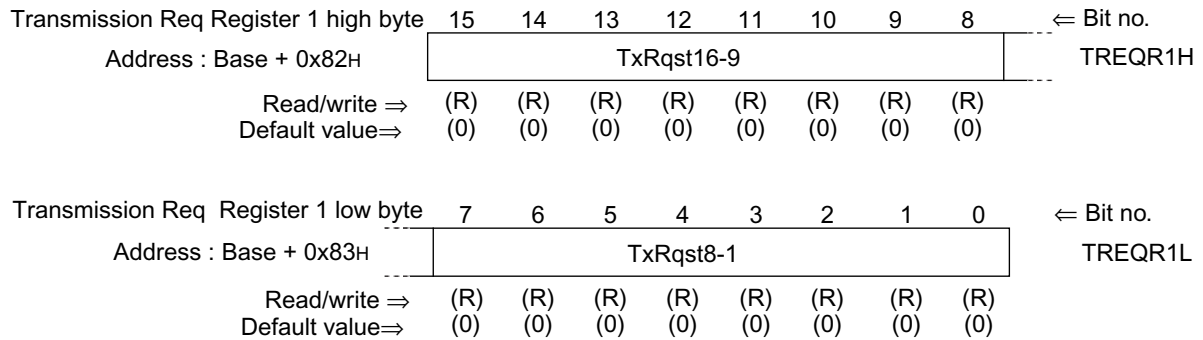


Figure 34.2-19 Transmission Request Register 1 (TREQR1)

**TxRqst32-1** Transmission Request Bits (of all Message Objects)

- 0 This Message Object is not waiting for transmission.
 1 The transmission of this Message Object is requested and is not yet done.

These registers hold the **TxRqst** bits of the 32 Message Objects. By reading out the **TxRqst** bits, the CPU can check for which Message Object a Transmission Request is pending. The **TxRqst** bit of a specific Message Object can be set/reset by the CPU via the IFx Message Interface Registers or by the Message Handler after reception of a Remote Frame or after a successful transmission.

Note: When the lowest priority message buffer is used for transmission, setting **TxRqst** = "0" may cause a delay of transmission, when **TxRqst** is set to "1" again.

Depending on the exact time when **TxRqst** was set to 0, the message may not be transmitted immediately after setting **TxRqst** = "1", but after any of the following events:

1. there is activity ongoing on the CANbus
2. a transmission request is issued on another message object
3. the CANbus is initialized by the INIT bit

In general, there is no need to cancel an ongoing transmission by setting **TxRqst** = 0. If the content of a message object needs to be changed while **TxRqst** = 1, it is sufficient to update the message object via the CPU interface registers (Identifier, **DLC**, **Data**, with **TxRqst** and **NewDat**, optionally **TxIE**). The updated content will be transmitted at the next opportunity.

If more than 32 message buffers are implemented, the following table gives an overview about the additional flags:

Table 34.2-2 Additional flags when more than 32 message buffers exist

		addr+0	addr+1	addr+2	addr+3
TREQR 4 & 3	TxRqst 64-33 (address 0x84)	TxRqst64-57	TxRqst56-49	TxRqst48-41	TxRqst40-33
TREQR 6 & 5	TxRqst 96-65 (address 0x88)	TxRqst96-89	TxRqst88-81	TxRqst80-73	TxRqst72-65

TREQR 8 & 7	TxRqst 128-97 (address 0x8C)	TxRqst128-121	TxRqst120-113	TxRqst112-105	TxRqst104-97
-------------	------------------------------	---------------	---------------	---------------	--------------

■ New Data Registers (NEWDT)

Figure 34.2-20 New Data Register 2 (NEWDT2)

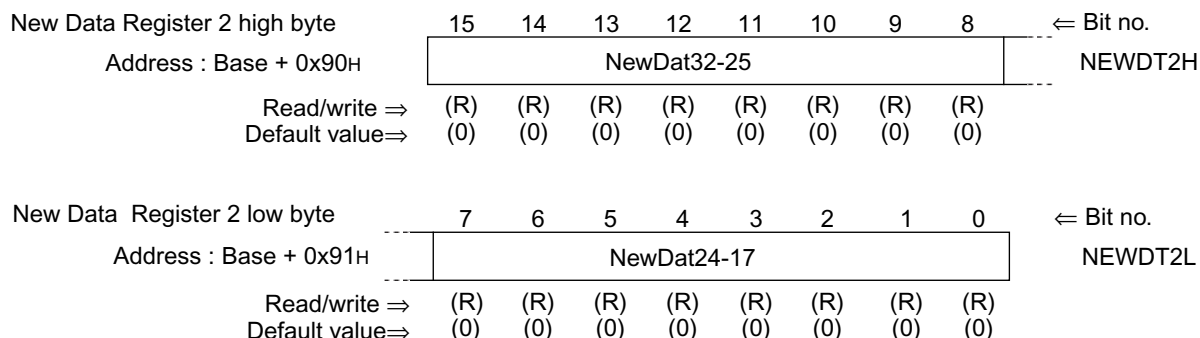
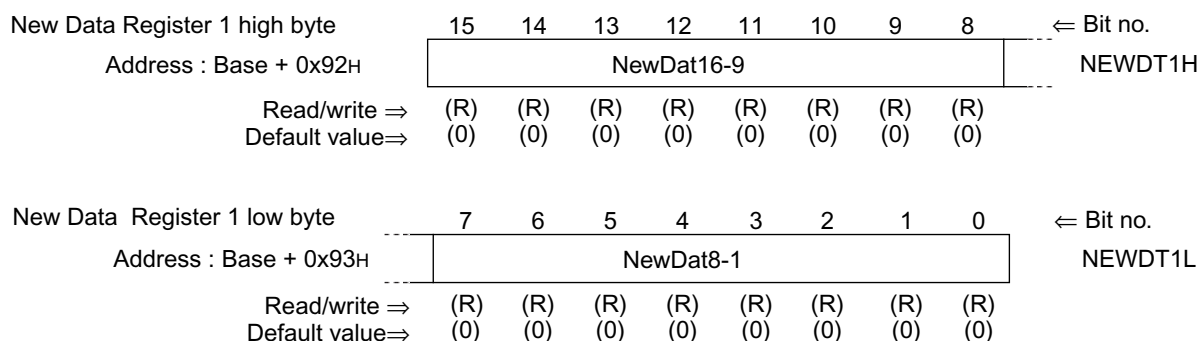


Figure 34.2-21 New Data Register 1 (NEWDT1)



NewDat32-1 New Data Bits (of all Message Objects)

- | | |
|---|--|
| 0 | No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the CPU. |
| 1 | The Message Handler or the CPU has written new data into the data portion of this Message Object. |

These registers hold the **NewDat** bits of the 32 Message Objects. By reading out the **NewDat** bits, the CPU can check for which Message Object the data portion was updated. The **NewDat** bit of a specific Message Object can be set/reset by the CPU via the IFx Message Interface Registers or by the Message Handler after reception of a Data Frame or after a successful transmission.

If more than 32 message buffers are implemented, the following table gives an overview about the additional flags:

Table 34.2-3 Additional flags when more than 32 message buffers exist

		addr+0	addr+1	addr+2	addr+3
NEWDT 4 & 3	NewDat 64-33 (address 0x94)	NewDat64-57	NewDat56-49	NewDat48-41	NewDat40-33
NEWDT 6 & 5	NewDat 96-65 (address 0x98)	NewDat96-89	NewDat88-81	NewDat80-73	NewDat72-65
NEWDT 8 & 7	NewDat 128-97 (address 0x9C)	NewDat128-121	NewDat120-113	NewDat112-105	NewDat104-97

■ Interrupt Pending Registers (INTPND)

Figure 34.2-22 Interrupt Pending Register 2 (INTPND2)

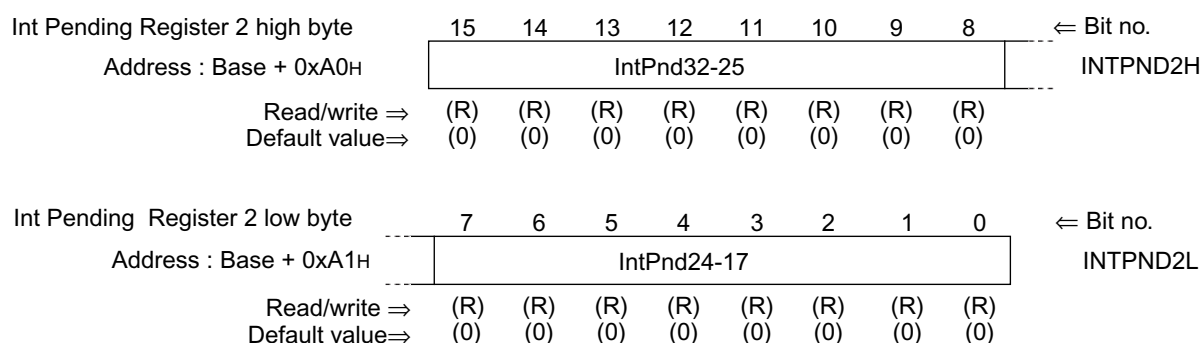
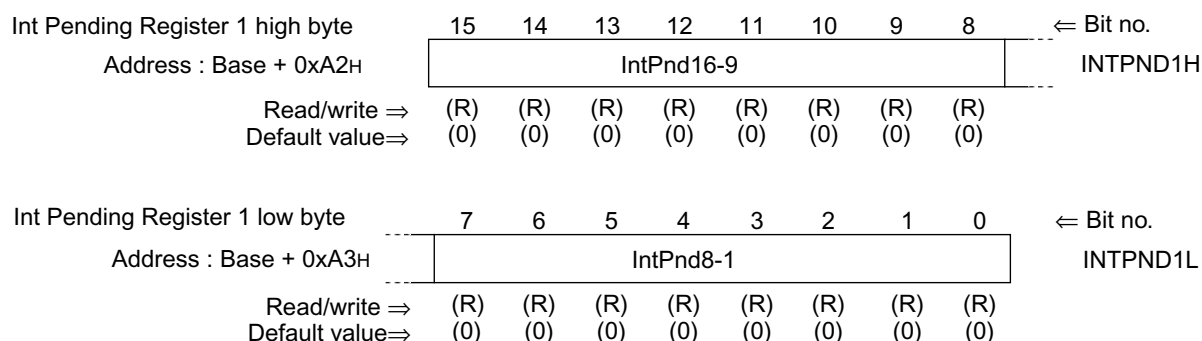


Figure 34.2-23 Interrupt Pending Register 1 (INTPND1)



IntPnd32-1 Interrupt Pending Bits (of all Message Objects)

- 0 This message object is not the source of an interrupt.
- 1 This message object is the source of an interrupt.

These registers hold the **IntPnd** bits of the 32 Message Objects. By reading out the **IntPnd** bits, the CPU can check for which Message Object an interrupt is pending. The **IntPnd** bit of a specific Message Object can be set/reset by the CPU via the IFx Message Interface Registers or by the Message Handler after reception or after a successful transmission of a frame. This will also affect the value of **IntId** in the Interrupt Register.

If more than 32 message buffers are implemented, the following table gives an overview about the additional flags:

Table 34.2-4 Additional flags when more than 32 message buffers exist

		addr+0	addr+1	addr+2	addr+3
INTPND 4 & 3	IntPnd 64-33 (address 0xA4)	IntPnd64-57	IntPnd56-49	IntPnd48-41	IntPnd40-33
INTPND 6 & 5	IntPnd 96-65 (address 0xA8)	IntPnd96-89	IntPnd88-81	IntPnd80-73	IntPnd72-65
INTPND 8 & 7	IntPd 128-97 (address 0xAC)	IntPnd128-121	IntPnd120-113	IntPnd112-105	IntPnd104-97

■ Message Valid Registers (MSGVAL)

Figure 34.2-24 Message Valid Register 2 (MSGVAL2)

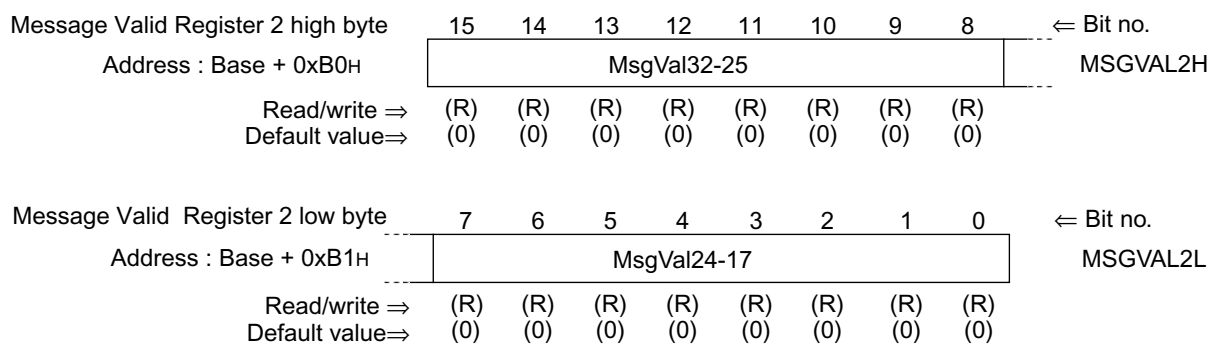
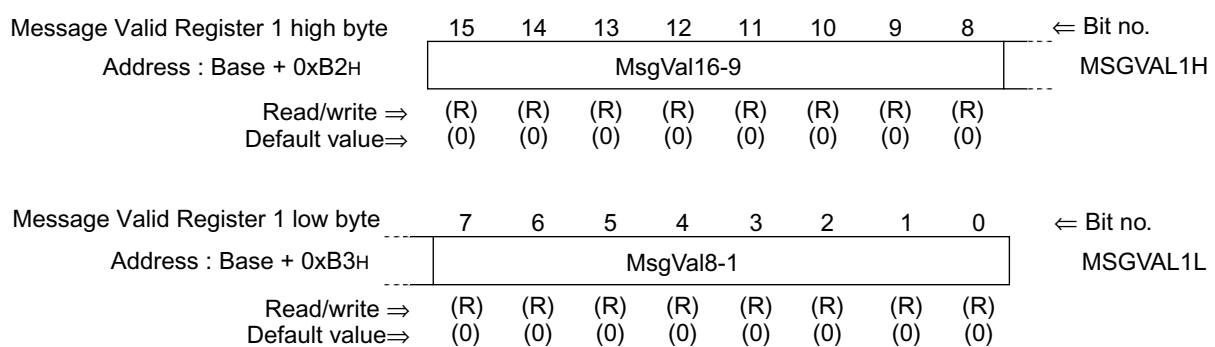


Figure 34.2-25 Message Valid Register 1 (MSGVAL1)



MsgVal32-1 Message Valid Bits (of all Message Objects)

- 0 This Message Object is ignored by the Message Handler.
- 1 This Message Object is configured and should be considered by the Message Handler.

These registers hold the **MsgVal** bits of the 32 Message Objects. By reading out the **MsgVal** bits, the CPU

MB91460 Series

can check which Message Object is valid. The **MsgVal** bit of a specific Message Object can be set/reset by the CPU via the IFx Message Interface Registers.

If more than 32 message buffers are implemented then the following table gives an overview about the additional flags:

Table 34.2-5 Additional flags when more than 32 message buffers exist

		addr+0	addr+1	addr+2	addr+3
MSGVAL 4 & 3	MsgVal 64-33 (address 0xB4)	MsgVal64-57	MsgVal56-49	MsgVal48-41	MsgVal40-33
MSGVAL 6 & 5	MsgVal 96-65 (address 0xB8)	MsgVal96-89	MsgVal88-81	MsgVal80-73	MsgVal72-65
MSGVAL 8 & 7	MsgVal 128-97 (address 0xBC)	MsgVal128-121	MsgVal120-113	MsgVal112-105	MsgVal104-97

34.3. Functional Description

This chapter provides an overview of the CAN module's operating modes and how to use them.

34.3.1 Software Initialization

The software initialization is started by setting the bit **Init** in the CAN Control Register, either by software or by a hardware reset, or by going Bus_Off.

While **Init** is set, all message transfers from and to the CAN bus are stopped, the status of the CAN bus output **TX** is recessive (HIGH). The counters of the EML are unchanged. Setting **Init** does not change any configuration register.

To initialize the CAN Controller, the CPU has to set up the Bit Timing Register and each Message Object. If a Message Object is not needed, it is sufficient to set its **MsgVal** bit to not valid. Otherwise, the whole Message Object has to be initialized.

Access to the Bit Timing Register and to the BRP Extension Register for the configuration of the bit timing is enabled when both bits **Init** and **CCE** in the CAN Control Register are set.

Resetting **Init** (by CPU only) finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (= Bus Idle) before it can take part in bus activities and starts the message transfer.

The initialization of the Message Objects is independent of **Init** and can be done on the fly, but the Message Objects should all be configured to particular identifiers or set to not valid before the BSP starts the message transfer.

To change the configuration of a Message Object during normal operation, the CPU has to start by setting **MsgVal** to not valid. When the configuration is completed, **MsgVal** is set to valid again.

34.3.2 CAN Message Transfer

Once the CAN is initialized and **Init** is reset to zero, the CAN's CAN Core synchronizes itself to the CAN bus and starts the message transfer.

Received messages are stored in their appropriate Message Objects if they pass the Message Handler's acceptance filtering. The whole message including all arbitration bits, DLC and eight data bytes is stored in the Message Object. If the Identifier Mask is used, the arbitration bits which are masked to "don't care" may be overwritten in the Message Object.

The CPU may read or write each message any time via the Interface Registers, the Message Handler guarantees data consistency in case of concurrent accesses.

Messages to be transmitted are updated by the CPU. If a permanent Message Object (arbitration and control bits set up during configuration) exists for the message, only the data bytes are updated and then **TxRqst** bit with **NewDat** bit are set to start the transmission. If several transmit messages are assigned to the same Message Object (when the number of Message Objects is not sufficient), the whole Message Object has to be configured before the transmission of this message is requested.

The transmission of any number of Message Objects may be requested at the same time, they are transmitted subsequently according to their internal priority. Messages may be updated or set to not valid any time, even when their requested transmission is still pending. The old data will be discarded when a message is updated before its pending transmission has started. Depending on the configuration of the Message Object, the transmission of a message may be requested autonomously by the reception of a remote frame with a matching identifier.

34.3.3 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898, 6.3.3 Recovery Management), the CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed. By default, this means for automatic retransmission is enabled. It can be disabled to enable the CAN to work within a Time Triggered CAN (TTCAN, see ISO11898-1) environment. The Disabled

Automatic Retransmission mode is enabled by setting the bit **DAR** in the CAN Control Register to one. In this operation mode the programmer has to consider the different behaviour of bits **TxRqst** and **NewDat** in the Control Registers of the Message Buffers:

When a transmission starts bit **TxRqst** of the respective Message Buffer is reset, while bit **NewDat** remains set.

When the transmission completed successfully bit **NewDat** is reset.

When a transmission failed (lost arbitration or error) bit **NewDat** remains set. To restart the transmission the CPU has to set **TxRqst** back to one.

When the host requests the transmission of several messages at the same time, only two of these messages will be transmitted. For all other requested transmit messages, the **TxRqst** bits will be reset, but no transmission will be started, **NewDat** and **IntPnd** will be left unchanged. For the two messages that are transmitted, the **TxRqst** and **NewDat** bits will be reset and, if enabled by **TxIE**, **IntPnd** will be set.

34.3.4 Test Mode

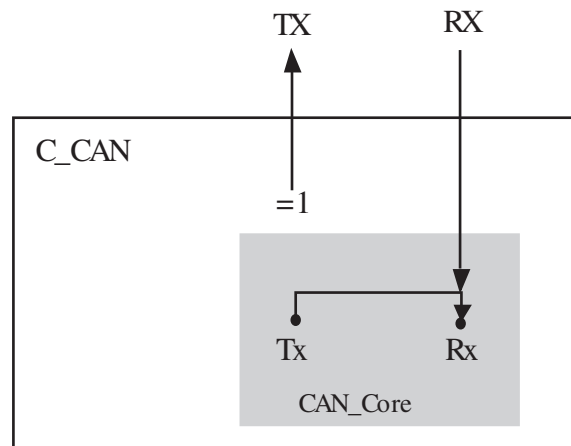
The Test Mode is entered by setting bit **Test** in the CAN Control Register to one. In Test Mode the bits **Tx1**, **Tx0**, **LBack**, **Silent** and **Basic** in the Test Register are writable. Bit **Rx** monitor the state of pin **RX** and therefore is only readable. All Test Register functions are disabled when bit **Test** is reset to zero.

34.3.5 Silent Mode

The CAN Core can be set in Silent Mode by programming the Test Register bit **Silent** to one.

In Silent Mode, the CAN is able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and it cannot start a transmission. If the CAN Core is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN Core monitors this dominant bit, although the CAN bus may remain in recessive state. The Silent Mode can be used to analyse the traffic on a CAN bus without affecting it by the transmission of dominant bits (Acknowledge Bits, Error Frames). Figure 34.3-1 shows the connection of signals **TX** and **RX** to the CAN Core in Silent Mode.

Figure 34.3-1 CAN Core in Silent Mode



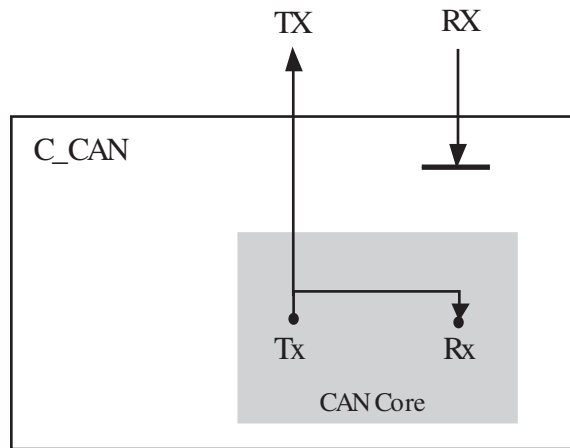
In ISO 11898-1, the Silent Mode is called the Bus Monitoring Mode.

34.3.6 Loop Back Mode

The CAN Core can be set in Loop Back Mode by programming the Test Register bit **LBack** to one. In Loop Back Mode, the CAN Core treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a Receive Buffer. Figure 34.3-2 shows the connection of signals **TX** and

RX to the CAN Core in Loop Back Mode.

Figure 34.3-2 CAN Core in Loop Back Mode

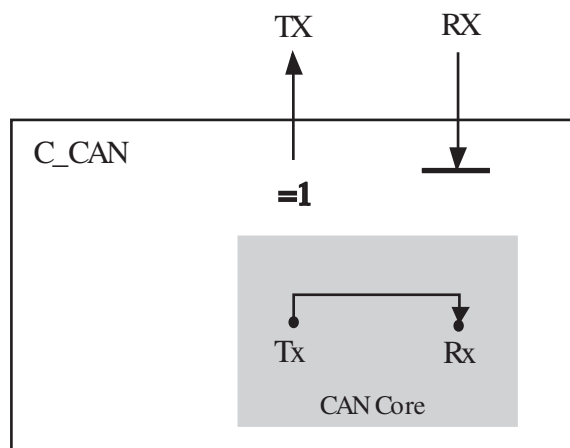


This mode is provided for self-test functions. To be independent from external stimulation, the CAN Core ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remot frame) in Loop Back Mode. In this mode the CAN Core performs an internal feedback from its Tx output to its Rx input. The actual value of the **RX** input pin is disregarded by the CAN Core. The transmitted messages can be monitored at the **TX** pin.

34.3.7 Loop Back combined with Silent Mode

It is also possible to combine Loop Back Mode and Silent Mode by programming bits **LBack** and **Silent** to one at the same time. This mode can be used for a “Hot Selftest”, meaning the CAN can be tested without affecting a running CAN system connected to the pins **TX** and **RX**. In this mode the **RX** pin is disconnected from the CAN Core and the **TX** pin is held recessive. Figure 34.3-3 shows the connection of signals **TX** and **RX** to the CAN Core in case of the combination of Loop Back Mode with Silent Mode.

Figure 34.3-3 CAN Core in Loop Back combined with Silent Mode



34.3.8 Basic Mode

The CAN Core can be set in Basic Mode by programming the Test Register bit **Basic** to one. In this mode the CAN module runs without the Message RAM.

The IF1 Registers are used as Transmit Buffer. The transmission of the contents of the IF1 Registers is requested by writing the **Busy** bit of the IF1 Command Request Register to '1'. The IF1 Registers are locked while the **Busy** bit is set. The **Busy** bit indicates that the transmission is pending.

As soon as the CAN bus is idle, the IF1 Registers are loaded into the shift register of the CAN Core and the transmission is started. When the transmission has completed, the **Busy** bit is reset and the locked IF1 Registers are released.

A pending transmission can be aborted at any time by resetting the **Busy** bit in the IF1 Command Request Register while the IF1 Registers are locked. If the CPU has reset the Busy bit, a possible retransmission in case of lost arbitration or in case of an error is disabled.

The IF2 Registers are used as Receive Buffer. After the reception of a message the content of the shift register is stored into the IF2 Registers, without any acceptance filtering.

Additionally, the actual contents of the shift register can be monitored during the message transfer. Each time a read Message Object is initiated by writing the **Busy** bit of the IF2 Command Request Register to '1', the content of the shift register is stored in the IF2 Registers.

In Basic Mode the evaluation of all Message Object related control and status bits and of the control bits of the IFx Command Mask Registers is turned off. The message number of the Command request registers is not evaluated. The **NewDat** and **MsgLst** bits of the IF2 Message Control Register retain their function, **DLC3-0** will show the received **DLC**, the other control bits will be read as '0'.

34.3.9 Software control of Pin TX

Four output functions are available for the CAN transmit pin **TX**. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor CAN_Core's bit timing and it can drive constant dominant or recessive values. The last two functions, combined with the readable CAN receive pin **RX**, can be used to check the CAN bus' physical layer.

The output mode of pin **TX** is selected by programming the Test Register bits **Tx1** and **Tx0** as described in section [“CAN Protocol Related Registers” on page 837](#)

The three test functions for pin **TX** interfere with all CAN protocol functions. **TX** must be left in its default function when CAN message transfer or any of the test modes Loop Back Mode, Silent Mode, or Basic Mode are selected.

34.4. CAN Application

This section describes how to use the CAN module in the application

34.4.1 Management of Message Objects

The configuration of the Message Objects in the Message RAM will (with the exception of the bits **MsgVal**, **NewDat**, **IntPnd**, and **TxRqst**) not be affected by resetting the chip. All the Message Objects must be initialized by the CPU or they must be not valid (**MsgVal** = '0') and the bit timing must be configured before the CPU clears the **Init** bit in the CAN Control Register.

The configuration of a Message Object is done by programming Mask, Arbitration, Control and Data field of one of the two interface register sets to the desired values. By writing to the corresponding IFx Command Request Register, the IFx Message Buffer Registers are loaded into the addressed Message Object in the Message RAM.

When the **Init** bit in the CAN Control Register is cleared, the CAN Protocol Controller state machine of the CAN_Core and the Message Handler State Machine control the CAN's internal data flow. Received messages that pass the acceptance filtering are stored into the Message RAM, messages with pending transmission request are loaded into the CAN_Core's Shift Register and are transmitted via the CAN bus.

The CPU reads received messages and updates messages to be transmitted via the IFx Interface Registers. Depending on the configuration, the CPU is interrupted on certain CAN message and CAN error events.

34.4.2 Message Handler State Machine

The Message Handler controls the data transfer between the Rx/Tx Shift Register of the CAN Core, the Message RAM and the IFx Registers.

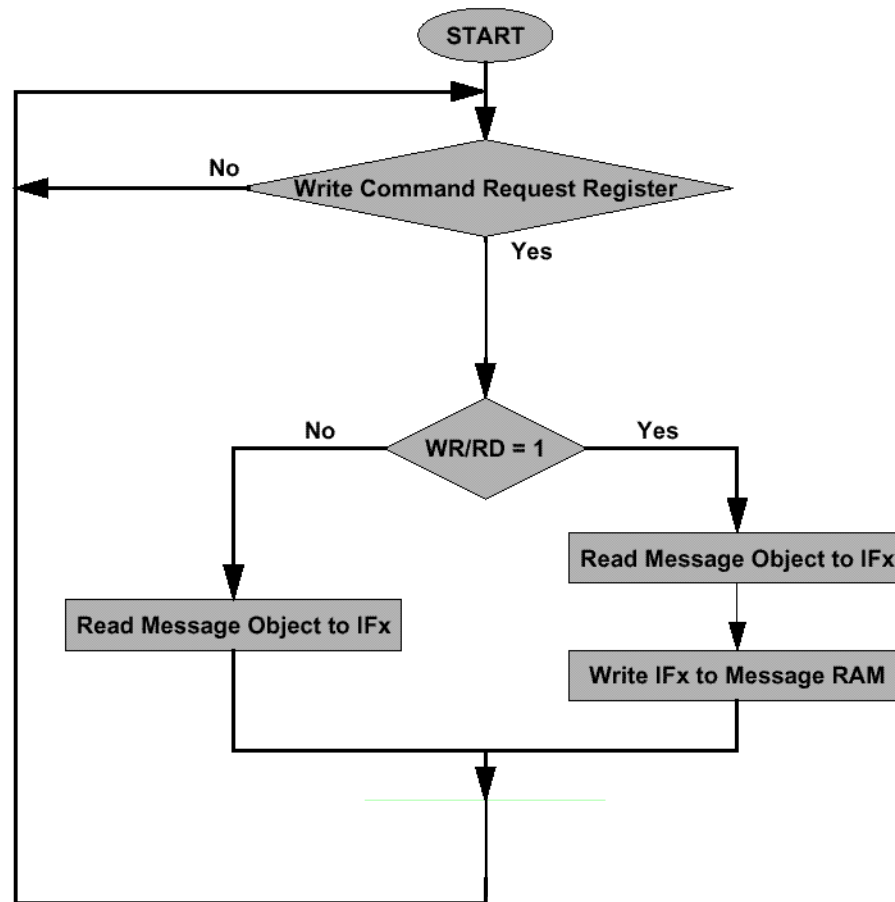
- The Message Handler FSM controls the following functions:
 - Data Transfer from IFx Registers to the Message RAM
 - Data Transfer from Message RAM to the IFx Registers
 - Data Transfer from Shift Register to the Message RAM
 - Data Transfer from Message RAM to Shift Register
 - Data Transfer from Shift Register to the Acceptance Filtering unit
 - Scanning of Message RAM for a matching Message Object
 - Handling of **TxRqst** flags.
 - Handling of interrupts.

34.4.3 Data Transfer from/to Message RAM

When the CPU initiates a data transfer between the IFx Registers and Message RAM, the Message Handler sets an internal busy signal which delays a consecutive access. After the transfer has completed, the busy signal is set back and the consecutive access is executed.

The respective Command Mask Register specifies whether a complete Message Object or only parts of it will be transferred. Due to the structure of the Message RAM it is not possible to write single bits/bytes of one Message Object, it is always necessary to write a complete Message Object into the Message RAM. Therefore the data transfer from the IFx Registers to the Message RAM requires a read-modify-write cycle. First parts of the Message Object that are not to be changes are read from the Message RAM and then the complete contents of the Message Buffer Registers are written into the Message Object.

Figure 34.4-1 Data Transfer between IFx Registers and Message RAM



After the partial write of a Message Object, that Message Buffer Registers that are not selected in the Command Mask Register will set to the actual contents of the selected Message Object.

After the partial read of a Message Object, that Message Buffer Registers that are not selected in the Command Mask Register will be left unchanged.

34.4.4 Transmission of Messages

If the shift register of the CAN Core cell is ready for loading and if there is no data transfer between the IFx Registers and Message RAM, the **MsgVal** bits in the Message Valid Register and the **TxRqst** bits in the Transmission Request Register are evaluated. The valid Message Object with the highest priority pending transmission request is loaded into the shift register by the Message Handler and the transmission is started. The Message Object's **NewDat** bit is reset.

After a successful transmission and if no new data was written to the Message Object (**NewDat** = '0') since the start of the transmission, the **TxRqst** bit will be reset. If **TxIE** is set, **IntPnd** will be set after a successful transmission. If the CAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

34.4.5 Acceptance Filtering of Received Messages

When the arbitration and control field (Identifier + IDE + RTR + DLC) of an incoming message is completely

shifted into the Rx/Tx Shift Register of the CAN Core, the Message Handler FSM starts the scanning of the Message RAM for a matching valid Message Object.

To scan the Message RAM for a matching Message Object, the Acceptance Filtering unit is loaded with the arbitration bits from the CAN Core shift register. Then the arbitration and mask fields (including **MsgVal**, **UMask**, **NewDat**, and **EoB**) of Message Object 1 are loaded into the Acceptance Filtering unit and compared with the arbitration field from the shift register. This is repeated with each following Message Object until a matching Message Object is found or until the end of the Message RAM is reached.

If a match occurs, the scanning is stopped and the Message Handler FSM proceeds depending on the type of frame (Data Frame or Remote Frame) received.

34.4.6 Reception of Data Frame

The Message Handler FSM stores the message from the CAN Core shift register into the respective Message Object in the Message RAM. Not only the data bytes, but all arbitration bits and the Data Length Code are stored into the corresponding Message Object. This is implemented to keep the data bytes connected with the identifier even if arbitration mask registers are used.

The **NewDat** bit is set to indicate that new data (not yet seen by the CPU) has been received. The CPU should reset **NewDat** when it reads the Message Object. If at the time of the reception the **NewDat** bit was already set, **MsgLst** is set to indicate that the previous data (supposedly not seen by the CPU) is lost. If the **RxIE** bit is set, the **IntPnd** bit is set, causing the Interrupt Register to point to this Message Object.

The **TxRqst** bit of this Message Object is reset to prevent the transmission of a Remote Frame, while the requested Data Frame has just been received.

34.4.7 Reception of Remote Frame

When a Remote Frame is received, three different configurations of the matching Message Object have to be considered:

1) **Dir** = '1' (direction = transmit), **RmtEn** = '1', **UMask** = '1' or '0'

At the reception of a matching Remote Frame, the **TxRqst** bit of this Message Object is set. The rest of the Message Object remains unchanged.

2) **Dir** = '1' (direction = transmit), **RmtEn** = '0', **UMask** = '0'

At the reception of a matching Remote Frame, the **TxRqst** bit of this Message Object remains unchanged; the Remote Frame is ignored.

3) **Dir** = '1' (direction = transmit), **RmtEn** = '0', **UMask** = '1'

At the reception of a matching Remote Frame, the **TxRqst** bit of this Message Object is reset. The arbitration and control field (Identifier + IDE + RTR + DLC) from the shift register is stored into the Message Object in the Message RAM and the **NewDat** bit of this Message Object is set. The data field of the Message Object remains unchanged; the Remote Frame is treated similar to a received Data Frame.

34.4.8 Receive / Transmit Priority

The receive/transmit priority for the Message Objects is attached to the message number. Message Object 1 has the highest priority, while Message Object 32 (the highest implemented message object number) has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding Message Object.

MB91460 Series

34.4.9 Configuration of a Transmit Object

Figure 34.4-2 shows how a Transmit Object should be initialised.

Figure 34.4-2 Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The Arbitration Registers (**ID28-0** and **Xtd** bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit Identifier ("Standard Frame") is used, it is programmed to **ID28 - ID18**, **ID17 - ID0** can then be disregarded.

If the **TxIE** bit is set, the **IntPnd** bit will be set after a successful transmission of the Message Object.

If the **RmtEn** bit is set, a matching received Remote Frame will cause the **TxRqst** bit to be set; the Remote Frame will autonomously be answered by a Data Frame.

The Data Registers (**DLC3-0**, **Data0-7**) are given by the application, **TxRqst** and **RmtEn** may not be set before the data is valid.

The Mask Registers (**Msk28-0**, **UMask**, **MXtd**, and **MDir** bits) may be used (**UMask**='1') to allow groups of Remote Frames with similar identifiers to set the **TxRqst** bit. For details see section 34.4.4 [Transmission of Messages \(Page No.867\)](#), handle with care. The **Dir** bit should not be masked.

34.4.10 Updating a Transmit Object

The CPU may update the data bytes of a Transmit Object any time via the IFx Interface registers, neither **MsgVal** nor **TxRqst** have to be reset before the update.

Note: When the lowest priority message buffer is used for transmission, setting **TxRqst** = "0" may cause a delay of transmission, when **TxRqst** is set to "1" again. Please refer to the [Note](#) at the description of **TxRqst** in section ["Transmission Request Registers \(TREQR\)" on page 856](#).

Even if only a part of the data bytes are to be updated, all four bytes of the corresponding IFx Data A Register or IFx Data B Register have to be valid before the content of that register is transferred to the Message Object. Either the CPU has to write all four bytes into the IFx Data Register or the Message Object is transferred to the IFx Data Register before the CPU writes the new data bytes.

When only the (eight) data bytes are updated, first 0x0087 is written to the Command Mask Register and then the number of the Message Object is written to the Command Request Register, concurrently updating the data bytes and setting **TxRqst**.

To prevent the reset of **TxRqst** at the end of a transmission that may already be in progress while the data is updated, **NewDat** has to be set together with **TxRqst**. For details see section 34.4.4 [Transmission of Messages \(Page No.867\)](#).

When **NewDat** is set together with **TxRqst**, **NewDat** will be reset as soon as the new transmission has started.

34.4.11 Configuration of a Receive Object

Figure 34.4-3 shows how a Transmit Object should be initialised.

Figure 34.4-3 Initialization of a Receive Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The Arbitration Registers (**ID28-0** and **Xtd** bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier ("Standard Frame") is used, it is programmed to **ID28 - ID18**, **ID17 - ID0** can then be disregarded. When a Data Frame with an 11-bit Identifier is received, **ID17 - ID0** will be set to '0'.

If the **RxIE** bit is set, the **IntPnd** bit will be set when a received Data Frame is accepted and stored in the Message Object.

The Data Length Code (**DLC3-0**) is given by the application. When the Message Handler stores a Data Frame in the Message Object, it will store the received Data Length Code and eight data bytes. If the Data Length Code is less than 8, the remaining bytes of the Message Object will be overwritten by **non specified values**.

The Mask Registers (**Msk28-0**, **UMask**, **MXtd**, and **MDir** bits) may be used (**UMask='1'**) to allow groups of Data Frames with similar identifiers to be accepted. For details see section 34.4.6 Reception of Data Frame (Page No.868). The **Dir** bit should not be masked in typical applications.

34.4.12 Handling of Received Messages

The CPU may read a received message any time via the IFx Interface registers, the data consistency is guaranteed by the Message Handler state machine.

Typically the CPU will write first 0x007F to the Command Mask Register and then the number of the Message Object to the Command Request Register. That combination will transfer the whole received message from the Message RAM into the Message Buffer Register. Additionally, the bits **NewDat** and **IntPnd** are cleared in the Message RAM (not in the Message Buffer).

If the Message Object uses masks for acceptance filtering, the arbitration bits show which of the matching messages have been received.

The actual value of **NewDat** shows whether a new message has been received since last time this Message Object was read. The actual value of **MsgLst** shows whether more than one message has been received since last time this Message Object was read. **MsgLst** will not be automatically reset.

By means of a Remote Frame, the CPU may request another CAN node to provide new data for a receive object. Setting the **TxRqst** bit of a receive object will cause the transmission of a Remote Frame with the receive object's identifier. This Remote Frame triggers the other CAN node to start the transmission of the matching Data Frame. If the matching Data Frame is received before the Remote Frame could be transmitted, the **TxRqst** bit is automatically reset.

34.4.13 Configuration of a FIFO Buffer

With the exception of the **EoB** bit, the configuration of Receive Objects belonging to a FIFO Buffer is the same as the configuration of a (single) Receive Object, see section 34.4.11 Configuration of a Receive Object (Page No.870).

To concatenate two or more Message Objects into a FIFO Buffer, the identifiers and masks (if used) of these Message Objects have to be programmed to matching values. Due to the implicit priority of the Message Objects, the Message Object with the lowest number will be the first Message Object of the FIFO Buffer. The **EoB** bit of all Message Objects of a FIFO Buffer except the last have to be programmed to zero. The **EoB** bits of the last Message Object of a FIFO Buffer is set to one, configuring it as the **End of the Block**.

34.4.14 Reception of Messages with FIFO Buffers

Received messages with identifiers matching to a FIFO Buffer are stored into a Message Object of this FIFO Buffer starting with the Message Object with the lowest message number.

When a message is stored into a Message Object of a FIFO Buffer the **NewDat** bit of this Message Object is set. By setting **NewDat** while **EoB** is 0 the Message Object is locked for further write accesses by the Message Handler until the CPU has written the **NewDat** bit back to 0.

Messages are stored into a FIFO Buffer until the last Message Object of this FIFO Buffer is reached. If none of the preceding Message Objects is released by writing **NewDat** to 0, all further messages for this FIFO Buffer will be written into the last Message Object of the FIFO Buffer and therefore overwrite previous messages.

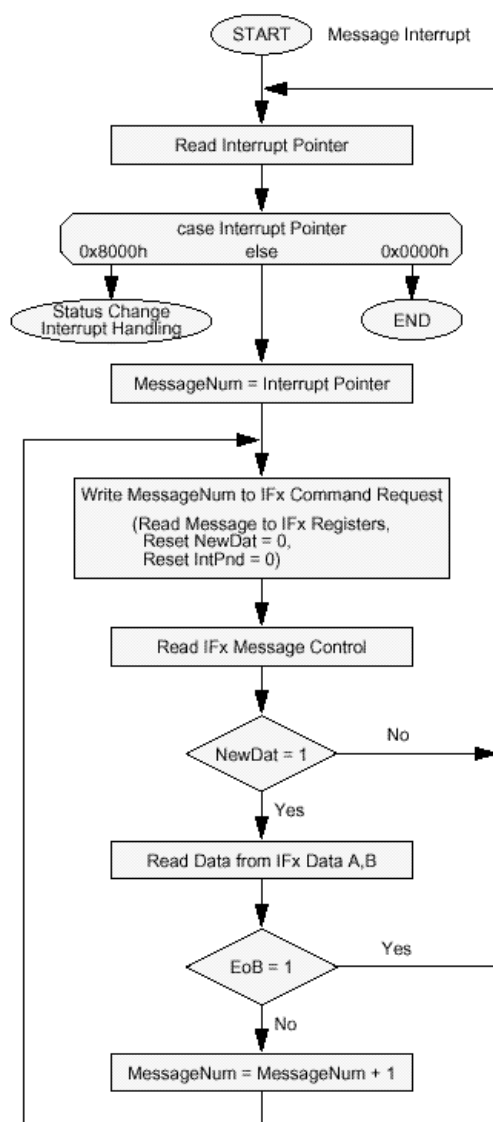
34.4.15 Reading from a FIFO Buffer

When the CPU transfers the contents of Message Object to the IFx Message Buffer registers by writing its number to the IFx Command Request Register, the corresponding Command Mask Register should be programmed the way that bits **NewDat** and **IntPnd** are reset to zero (**TxRqst/NewDat** = '1' and **ClrIntPnd** = '1'). The values of these bits in the Message Control Register always reflect the status before resetting the bits.

To assure the correct function of a FIFO Buffer, the CPU should read out the Message Objects starting at the FIFO Object with the lowest message number.

Figure 34.4-4 shows how a set of Message Objects which are concatenated to a FIFO Buffer can be handled by the CPU.

Figure 34.4-4 CPU Handling of a FIFO Buffer



34.4.16 Handling of Interrupts

If several interrupts are pending, the CAN Interrupt Register will point to the pending interrupt with the highest priority, disregarding their chronological order. An interrupt remains pending until the CPU has cleared it.

The Status Interrupt has the highest priority. Among the message interrupts, the Message Object's interrupt priority decreases with increasing message number.

A message interrupt is cleared by clearing the Message Object's IntPnd bit. The Status Interrupt is cleared by reading the Status Register.

The interrupt identifier **IntId** in the Interrupt Register indicates the cause of the interrupt. When no interrupt is pending, the register will hold the value zero. If the value of the Interrupt Register is different from zero, then there is an interrupt pending and, if **IE** is set, the interrupt line to the CPU is active. The interrupt line remains active until the Interrupt Register is back to value zero (the cause of the interrupt is reset) or until **IE** is reset.

The value 0x8000 indicates that an interrupt is pending because the CAN Core has updated (not necessarily changed) the Status Register (Error Interrupt or Status Interrupt). This interrupt has the highest priority. The CPU can update (reset) the status bits **RxOk**, **TxOk** and **LEC**, but a write access of the CPU to the Status Register can never generate or reset an interrupt.

All other values indicate that the source of the interrupt is one of the Message Objects, **IntId** points to the pending message interrupt with the highest interrupt priority.

The CPU controls whether a change of the Status Register may cause an interrupt (bits **EIE** and **SIE** in the CAN Control Register) and whether the interrupt line becomes active when the Interrupt Register is different from zero (bit **IE** in the CAN Control Register). The Interrupt Register will be updated even when **IE** is reset.

The CPU has two possibilities to follow the source of a message interrupt. First it can follow the **IntId** in the Interrupt Register and second it can poll the Interrupt Pending Register (see section [34.2.6 Message Handler Registers \(Page No.855\)](#)).

An interrupt service routine reading the message that is the source of the interrupt may read the message and reset the Message Object's **IntPnd** at the same time (through setting bit **CIP = 1** (clear interrupt pending) in the Command Mask Register). When **IntPnd** is cleared, the Interrupt Register will point to the next Message Object with a pending interrupt.

34.4.17 Bit Time and Bit Rate

The timing parameter of the bit time (i.e. the reciprocal of the bit rate) can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable, small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronising to the bit stream.

According to the CAN specification, the bit time is divided into four segments (see [Figure 34.4-5](#)). The Synchronization Segment, the Propagation Time Segment, the Phase Buffer Segment 1, and the Phase Buffer Segment 2. Each segment consists of a specific, programmable number of time quanta (see [Table 34.4-1 on page 874](#)). The length of the time quantum (t_q), which is the basic time unit of the bit time, is defined by the CAN controller's system clock f_{sys} and the Baud Rate Prescaler (BRP) : $t_q = BRP / f_{sys}$. The CAN's system clock f_{sys} is the frequency of its **CLKCAN** input.

The Synchronization Segment Sync_Seg is that part of the bit time where edges of the CAN bus level are expected to occur; the distance between an edge that occurs outside of Sync_Seg and the Sync_Seg is called the phase error of that edge. The Propagation Time Segment Prop_Seg is intended to compensate for the physical delay times within the CAN network. The Phase Buffer Segments Phase_Seg1 and Phase_Seg2 surround the Sample Point. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Figure 34.4-5 Bit Timing

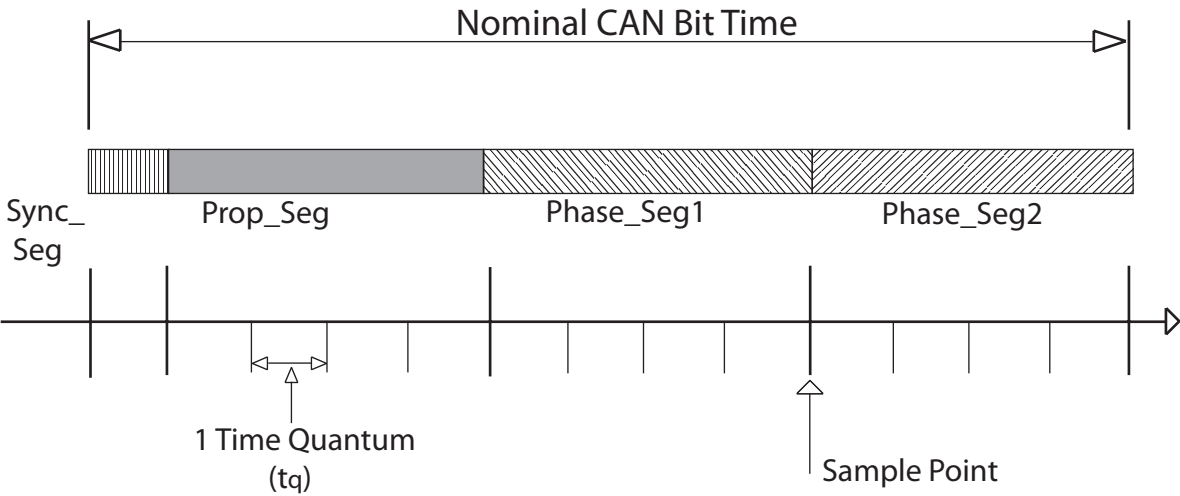


Table 34.4-1 Parameters of the CAN Bit Time

Parameter	Range	Remark
BRP	[1..32]	defines the length of the time quantum tq
Sync_Seg	1 tq	fixed length, synchronization of us into system clock
Prop_Seg	[1..8] tq	compensates for the physical delay times
Phase_Seg1	[1..8] tq	may be lengthened temporarily by synchronization
Phase_Seg2	[1..8] tq	may be shortened temporarily by synchronization
SJW	[1..4] tq	may not be longer than either Phase Buffer Segment

Note : This table describes the minimum programmable ranges required by the CAN protocol.

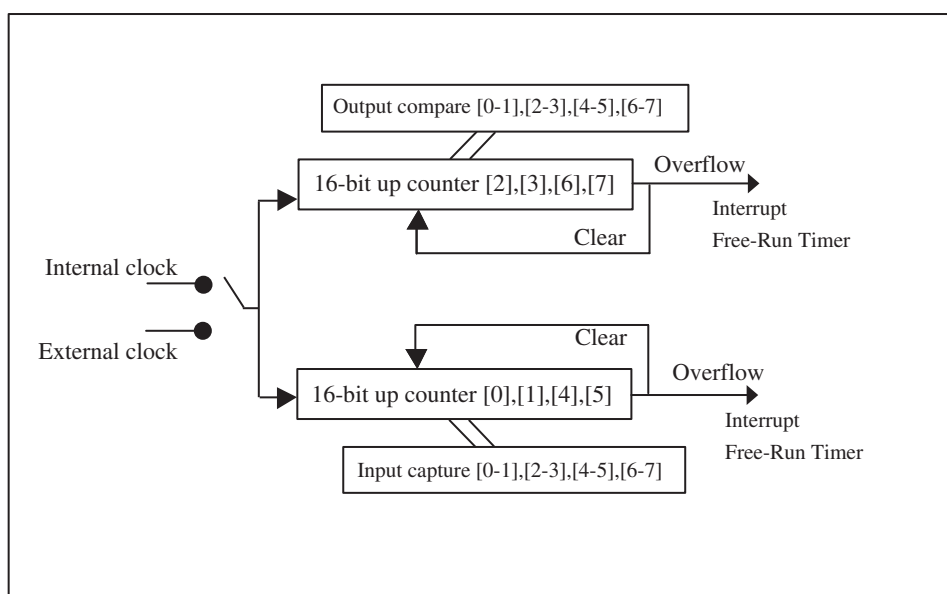
Chapter 35 Free-Run Timer (FRT)

35.1. Overview

The Free-Run timer consists of a 16-bit timer (up counter) and control circuits.

The Free-Run timer can be used by the input capture or by the output compare.

Figure 35.1-1 Block diagram of Free-Run Timer



35.2. Features

- Format: 16-bit up counter
- Quantity: 8 (Free-Run timer 0 up to Free-Run timer 7)
- Quantity: 12 (Free-Run timer 0 up to Free-Run timer 11) on MB91FV460B.

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

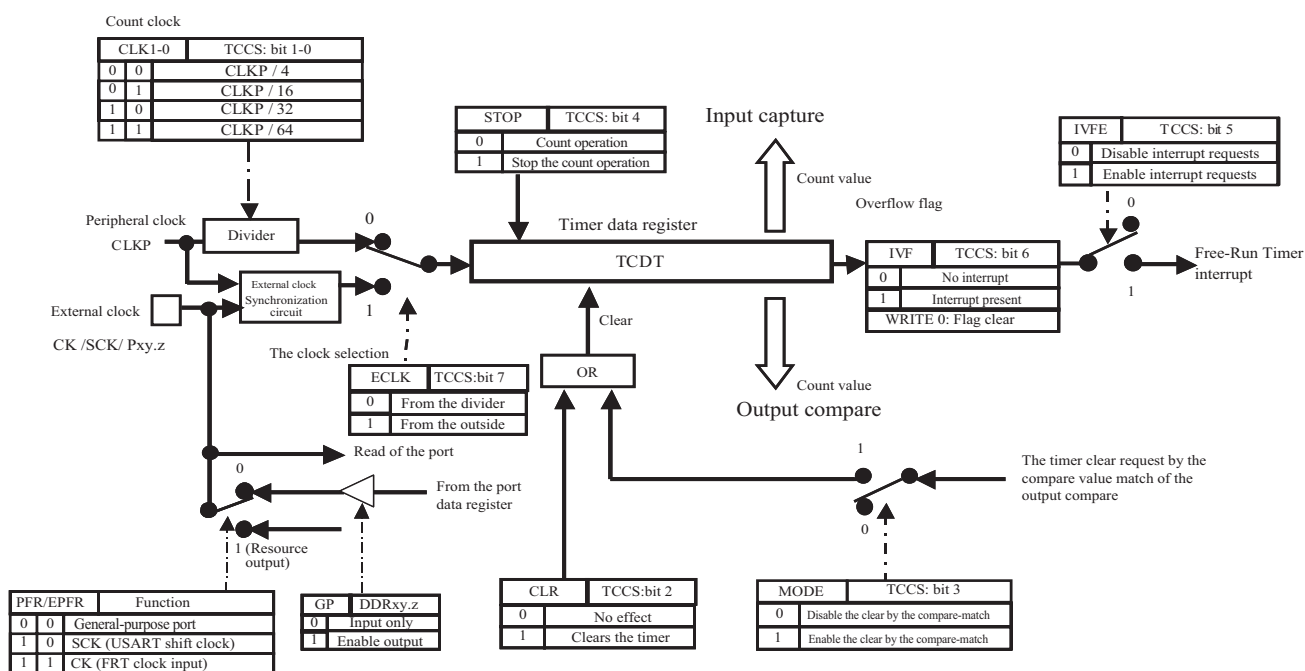
- Clock source: 4 internal clocks (1/4, 1/16, 1/32, and 1/64 of CLKP)
- External clock (CK pin)
- Clear factor of the counter:
 - Software
 - Reset
 - Compare-match (match of the compare-register value and the count value of the Free-Run timer)
- Operation start/stop: operations can be started/stopped by software.
- Interrupt:
 - Overflow interrupt
 - An interrupt generated when the compare clear register value and the count value of the Free-Run timer match.
- Count value: Readable/writable (write is only possible when the counting stops)
- Others: Operates from immediately after reset.
- Free-Run Timer and ICU/OCU:

- Free-Run timer 0 can be used by input capture unit 0/1
- Free-Run timer 1 can be used by input capture unit 2/3
- Free-Run timer 2 can be used by output compare unit 0/1
- Free-Run timer 3 can be used by output compare unit 2/3
- Free-Run timer 4 can be used by input capture unit 4/5
- Free-Run timer 5 can be used by input capture unit 6/7
- Free-Run timer 6 can be used by output compare unit 4/5
- Free-Run timer 7 can be used by output compare unit 6/7
- Free-Run timer 8 can be used by input capture unit 8/9

35.3. Configuration Diagram

Figure 35.3-1 Configuration Diagram of Free-Run Timer

Free-Run Timer



Notes: When using the input/output (SCK), the external clock (CK) cannot be used because the port is shared.

Figure 35.3-2 Register List

Free-run Timer 0

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
001F0H		T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	TCDT0	(Free-run Timer 0)

	Bit	7	6	5	4	3	2	1	0									
001F3H		ECLK	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0	TCCS0	(Free-run timer control 0)							
00D95H		---	---	---	---	---	SCK	---	---	PFR21	(Port function)							
00DD5H		---	---	---	---	---	CK	---	---	EPFR21	(Extra port function)							
0044CH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR12	(Interrupt level)							
0FFF5CH		32Bits																(Interrupt vector #40)

* See the chapter of "Interrupt Control" about ICR register and interrupt vectors.

Note: See “[Chapter 24 Interrupt Control \(Page No.429\)](#)” about ICR register and interrupt vectors.

35.4. Registers

35.4.1 TCCS: Timer Control Register

A register for controlling the operation of the Free-Run timer.

- **TCCS0** (Free-Run timer 0): Address 01F3h (access: Byte, Half-word, Word)
- **TCCS1** (Free-Run timer 1): Address 01F7h (access: Byte, Half-word, Word)
- **TCCS2** (Free-Run timer 2): Address 01FBh (access: Byte, Half-word, Word)
- **TCCS3** (Free-Run timer 3): Address 01FFh (access: Byte, Half-word, Word)
- **TCCS4** (Free-Run timer 4): Address 02F3h (access: Byte, Half-word, Word)
- **TCCS5** (Free-Run timer 5): Address 02F7h (access: Byte, Half-word, Word)
- **TCCS6** (Free-Run timer 6): Address 02FBh (access: Byte, Half-word, Word)
- **TCCS7** (Free-Run timer 7): Address 02FFh (access: Byte, Half-word, Word)
- **TCCS8** (Free-Run timer 8): Address 060Bh (access: Byte, Half-word, Word)
- **TCCS9** (Free-Run timer 9): Address 060Fh (access: Byte, Half-word, Word)
- **TCCS10** (Free-Run timer 10): Address 0613h (access: Byte, Half-word, Word)
- **TCCS11** (Free-Run timer 11): Address 0617h (access: Byte, Half-word, Word)

7	6	5	4	3	2	1	0	bit
ECLK	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0	
0	0	0	0	0	0	0	0	Initial value
R/W	R (RM1), W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit7: Select the count clock

ECLK	Select the count clock
0	Internal clock (the peripheral clock divided by n)
1	External clock (CK pin)

- The setting of the count clock selection bit should be changed only when other peripheral modules (the output compare, input capture, etc.) using the output of the Free-Run timer are stopped.
- When using the external clock, the period of the external clock must be more than double of the peripheral clock (CLKP). When using the output compare, in order to allow the compare-match output and interrupt generation, the external clock input of at least 1 clock is required after the compare-match.
- bit6: Interrupt flag

IVF	Status	
	Read	Write
0	No interrupt present	Clear the flag (IVF).
1	Interrupt present (Overflow or compare-match)	No effect on operation

- When the count value of the Free-Run timer overflows, or the clear mode bit (MODE) is “1”, the interrupt flag is set to “1” if the count values of the Free-Run timer and the compare register (OCCP) match and the counter is cleared.
- To enable the interrupt request, the interrupt enable bit must be set to do so (IVFE=“1”).
- When the interrupt flag is set to “1”, and “0” is written at the same time, the interrupt flag is set to “1”. (Setting the flag has priority.)
- bit5: Enable interrupt requests

IVFE	Operation
------	-----------

MB91460 Series

0	Disable interrupt requests
1	Enable interrupt requests

- When the interrupt request enable bit is set to “1”, the interrupt request (IVF) is enabled.
- bit4: Stop counting

STOP	Operation
0	Enable counting
1	Disable count (stop)

- When the count stop bit is set to “1”, the Free-Run timer stops.
- When the output compare is being used, if the Free-Run timer stops, the output compare also stops.
- bit3: Clear mode

MODE	Clear mode
0	Clear the Free-Run timer by the reset and the clear bit (CLR).
1	Clear the free-run timer by the match with the reset, the clear bit (CLR), and the compare register value of the output compare (OCCP).

- Set the clear mode of the Free-Run timer.
- If the clear mode bit is set to “1”, when the count value of the Free-Run timer and the compare-register value (OCCP) match, the count value of the Free-Run timer is cleared to “0000h”.
- The reset and writing “1” to the clear bit (CLR) cause to clear the count value of the Free-Run timer to “0000h”, regardless of the setting of the clear mode bit.
- The count value of the Free-Run timer is only cleared when the Free-Run timer is running. When the Free-Run timer is stopped, clear it by writing “0000h” to the timer data register (TCDT).
- bit2: Clear

CLR	Operation
0	No effect on operation
1	Clear the Free-Run timer.

- When the clear bit is set to “1”, the count value of the Free-Run timer is cleared to “0000h”. The clear bit is read as “1” until the Free-Run timer is completely cleared.
When the Free-Run timer is completely cleared, the clear bit is also cleared to “0”.
- When the clear operation of the Free-Run timer and writing “1” to the clear bit occurs at the same time, the clear bit keeps “1”, and after the next time the Free-Run timer is cleared, it is cleared.
- bit1-bit0: Count clock division ratio selection (when the internal clock is selected)

CLK1	CLK0	The division ratio of the count clock
0	0	Peripheral clock (CLKP) divided by 4
0	1	Peripheral clock (CLKP) divided by 16
1	0	Peripheral clock (CLKP) divided by 32
1	1	Peripheral clock (CLKP) divided by 64

- Select the division ratio of the count clock of the Free-Run timer.
- Change the division ratio when the setting of the count clock division ratio selection bit is changed. When the internal clock is selected as the count clock of the Free-Run timer (count clock selection bit ECLK=“0”), change the setting when other peripheral modules (output compare, input capture, etc.) using the output of the Free-Run timer are stopped.

35.4.2 TCDT: Timer Data Register

This register can read 16-bit Free-Run timer count values.

- **TCDT0 (Free-Run timer 0): Address 01F0h (access: Half-word, Word)**
- **TCDT1 (Free-Run timer 1): Address 01F4h (access: Half-word, Word)**
- **TCDT2 (Free-Run timer 2): Address 01F8h (access: Half-word, Word)**
- **TCDT3 (Free-Run timer 3): Address 01FCh (access: Half-word, Word)**
- **TCDT4 (Free-Run timer 4): Address 02F0h (access: Half-word, Word)**
- **TCDT5 (Free-Run timer 5): Address 02F4h (access: Half-word, Word)**
- **TCDT6 (Free-Run timer 6): Address 02F8h (access: Half-word, Word)**
- **TCDT7 (Free-Run timer 7): Address 02FCh (access: Half-word, Word)**
- **TCDT8 (Free-Run timer 8): Address 0608h (access: Half-word, Word)**
- **TCDT9 (Free-Run timer 9): Address 060Ch (access: Half-word, Word)**
- **TCDT10 (Free-Run timer 10): Address 0610h (access: Half-word, Word)**
- **TCDT11 (Free-Run timer 11): Address 0614h (access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
T15	T14	T13	T12	T11	T10	T9	T8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
T7	T6	T5	T4	T3	T2	T1	T0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(About attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- When the timer data register is read, the count value of the Free-Run timer is also obtained.
- By writing to the timer data register, the timer value can be written in the Free-Run timer. When it is written, make sure that the Free-Run timer is in the idle state (the count stop bit (TCCS.STOP= 1)).

MB91460 Series**35.5. Addresses and Connections**

The Free-Run Timers 8 - 11 are added to serve the ICU module 8/9 and to give more FRT resources. These FRTs (8-11) and ICU modules (8/9) are only available on MB91FV460B and later devices. The following table shows addresses and connections for all Free-Run Timers.

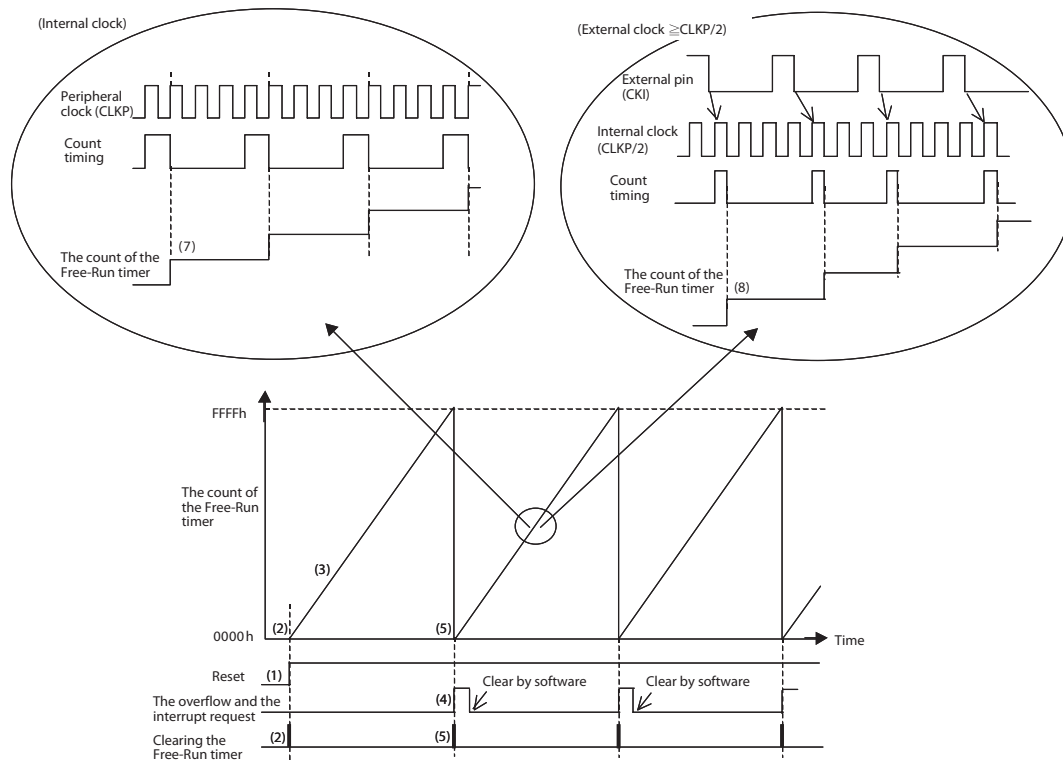
Address	Free-Run Timer	Trigger Input Port	FRT Interrupt Nr.	FRT DMA Resource Nr.	Timer value connected to...
0001F0 _H	FRT 0	GP21_2	40	40	Input Capture Unit 0/1
0001F4 _H	FRT 1	GP21_6	41	41	Input Capture Unit 2/3
0001F8 _H	FRT 2	GP20_2	42	42	Output Compare Unit 0/1
0001FC _H	FRT 3	GP20_6	43	43	Output Compare Unit 2/3
0002F0 _H	FRT 4	GP19_2	44	44	Input Capture Unit 4/5
0002F4 _H	FRT 5	GP19_6	45	45	Input Capture Unit 6/7
0002F8 _H	FRT 6	GP18_2	46	46	Output Compare Unit 4/5
0002FC _H	FRT 7	GP18_6	47	47	Output Compare Unit 6/7
000608 _H	FRT 8	GP21_2 * ¹	40	176	Input Capture Unit 8/9
00060C _H	FRT 9	GP21_6 * ²	41	177	-
000610 _H	FRT 10	GP20_2 * ³	42	178	-
000614 _H	FRT 11	GP20_6 * ⁴	43	179	-

1. Input and interrupt shared with FRT 0
2. Input and interrupt shared with FRT 1
3. Input and interrupt shared with FRT 2
4. Input and interrupt shared with FRT 3

35.6. Operation

35.6.1 Count Operation of the Free-Run Timer

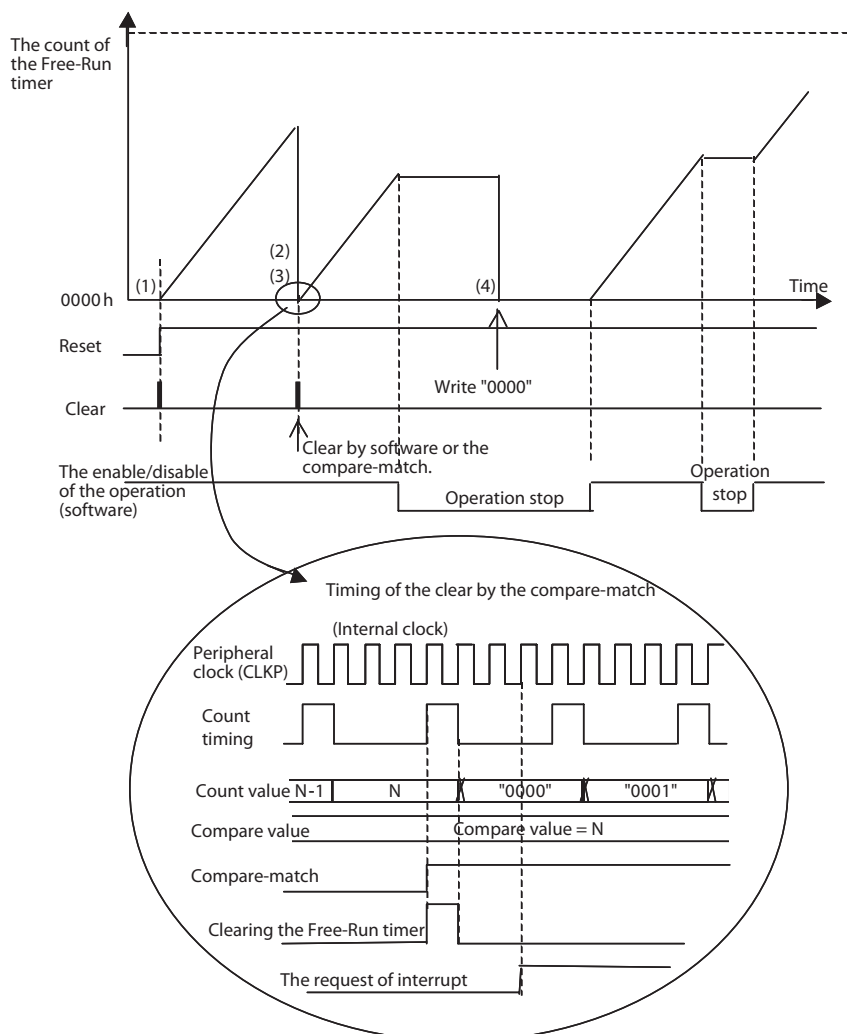
Figure 35.6-1 Count Operation of the Free-Run Timer



- (1) Reset
- (2) Clearing of the Free-Run timer by reset. (Count value "0000")
- (3) Count-up of the Free-Run timer
- (4) Overflow and interrupt of the Free-Run timer.
- (5) Clearing of the Free-Run timer by overflow. (Count value "0000")
- (6) Repeat (3) to (5)
- (7) The Free-Run timer counts up at the count clock (the internal clock divided by n).
- (8) The Free-Run timer counts up at the count clock (the external clock synchronized with the internal clock).

35.6.2 Various Clear Operations of the Free-Run Timer

Figure 35.6-2 Various Clear Operations of the Free-Run Timer



Clear operations of the Free-Run timer (4 types)

- (1) Reset
- (2) Clear by software
- (3) Clear by the compare-match
- (4) Writing "0000"

35.7. Settings

Table 35.7-1 Setting Required in Order to Use the Free-Run Timer

Setting	Setting Registers	Setting Procedures *
Setting of the initialization conditions of the timer	Timer control register (TCCS0-TCCS11)	See 35.8.4
Setting of the count clock Selection of the internal clock		See 35.8.1
Selection of the external clock		See 35.8.2
Start the count operation		See 35.8.3
In the case of the external clock Set the clock input pin (CK) as the input.	Port function register (PFRxy.z) Extra port function register (EPFRxy.z)	See 35.8.2

*: For the setting procedure, refer to the section indicated by the number.

Table 35.7-2 Setting Required to Enable the Free-Run Timer Interrupt

Setting	Setting Registers	Setting Procedures *
Setting of the Free-Run timer interrupt vector, and the Free-Run timer interrupt level	See “ Chapter 24 Interrupt Control (Page No.429) ”.	See 35.8.5
Setting of the Free-Run timer interrupt Clearing interrupt flags Enabling interrupt requests	Timer control register (TCCS0-TCCS11)	See 35.8.7

*: For the setting procedure, refer to the section indicated by the number.

Table 35.7-3 Setting Required to Stop the Free-Run Timer

Setting	Setting Registers	Setting Procedures *
Setting of the Free-Run timer stop bit	Timer control register (TCCS0-TCCS11)	See 35.8.8

*: For the setting procedure, refer to the section indicated by the number.

MB91460 Series**35.8. Q & A****35.8.1 What are the types of the internal clock, and how to select?**

There are 4 types of internal clocks, and these are set by the clock selection bit (TCCS.ECLK) and the count clock bit (TCCS.CLK [1:0]).

Internal clock	Setting		Count period	
	Clock selection Bit (ECLK)	Count clock bit (CLK [1:0])	CLKP = 32MHz	CLKP = 16MHz
To select CLKP/4	Set to "0"	Sets to "00"	125 ns	250 ns
To select CLKP/16	Set to "0"	Set to "01"	0.5 μ s	1 μ s
To select CLKP/32	Set to "0"	Set to "10"	1 μ s	2 μ s
To select CLKP/64	Set to "0"	Set to "11"	2 μ s	4 μ s

35.8.2 How to select the external clock

Set with clock selection bits (TCCS.ECLK), data direction bits, and (extra) port function bits.

To use the external clock input	Setting			Pins	Count Cycle
Free-Run Timer 0-11	The clock selection bit (ECLK) to "1"	The port function bit (PFRxy.z) to "1"	The extra port function bit (EPFRxy.z) to "1"	CK0-CK11	Over 2/CLKP

35.8.3 How to enable / disable the count operation of the Free-Run timer

Set with count operation bits (TCCS.STOP).

Operation	Count operation bit (STOP)
To enable the Free-Run timer	Set to "0"
To stop the Free-Run timer	Set to "1"

35.8.4 How to clear the Free-Run timer

The Free-Run timer can be cleared by performing the following operations:

- Set with clear bit (TCCS.CLR).

Operation	Clear bit (CLR)
To clear the Free-Run timer	Write "1"

- How to clear the Free-Run timer when the Free-Run timer value and the compare-register value match
Set with the timer initialization condition bit (TCCS.MODE).

Operation	Timer initialization condition bit (MODE)
To clear the Free-Run timer at the compare-match	Set to "1"

The setting of the output compare is also required. (See "[Chapter 37 Output Compare Unit \(OCU\) \(Page No.903\)](#)".)

- Reset.

When a reset occurs (the INITX pin input, the watchdog reset, the software reset), the Free-Run timer is cleared.

- Write "0000_H" while the Free-Run timer is stopped.

The count value will be set to "0000_H", when "0000_H" is written while the Free-Run timer is stopped.

- With the overflow of the Free-Run timer, the count value returns to "0000_H".

35.8.5 What interrupt registers are used?

Setting of the Free-Run timer interrupt vector and the Free-Run timer interrupt level

The relationship among the Free-Run timer number, interrupt levels and vectors is shown in the table below.

See "[Chapter 24 Interrupt Control \(Page No.429\)](#)" about the details of interrupt levels and interrupt vectors.

Number	Interrupt Vectors (default)	Interrupt level setting bits (ICR[4:0])
Free-Run Timer 0	#40 Address: 0FFF5Ch	Interrupt level register (ICR12) Address: 0044Ch
Free-Run Timer 1	#41 Address: 0FFF58h	
Free-Run Timer 2	#42 Address: 0FFF54h	Interrupt level register (ICR13) Address: 0044Dh
Free-Run Timer 3	#43 Address: 0FFF50h	
Free-Run Timer 4	#44 Address: 0FFF4Ch	Interrupt level register (ICR14) Address: 0044Eh
Free-Run Timer 5	#45 Address: 0FFF48h	
Free-Run Timer 6	#46 Address: 0FFF44h	Interrupt level register (ICR15) Address: 0044Fh
Free-Run Timer 7	#47 Address: 0FFF40h	
Free-Run Timer 8 (Shared with FRT0)	#40 Address: 0FFF5Ch	Interrupt level register (ICR12) Address: 0044Ch
Free-Run Timer 9 (Shared with FRT1)	#41 Address: 0FFF58h	
Free-Run Timer 10 (Shared with FRT2)	#42 Address: 0FFF54h	Interrupt level register (ICR13) Address: 0044Dh
Free-Run Timer 11 (Shared with FRT3)	#43 Address: 0FFF50h	

MB91460 Series

Since the interrupt request flag (TCCS.IVF) is not cleared automatically, make sure to clear it with software before returning from the interrupt process. (Write "0" in the IVF bit)

35.8.6 Interrupt Types

There is only one type of interrupt, and it is generated at the overflow of the free-run timer. (Selection is not required)

35.8.7 How to enable interrupts

Enable interrupt request bit, interrupt flag

Use interrupt request enable bit (TCCS.IVFE) to enable interrupts.

	Interrupt request enable bit (IVFE)
Disable interrupt requests	Set to "0"
Enable interrupt requests	Set to "1"

Use interrupt flag (TCCS.IVF) to clear interrupt requests.

	Interrupt flag (IVF)
Clear interrupt requests	Write "0"

35.8.8 How to stop the Free-Run timer

Set with count operation bit (TCCS.STOP).

See ["35.8.3 How to enable / disable the count operation of the Free-Run timer \(Page No.885\)"](#).

35.8.9 How are the Free-Run timer assigned to ICU and OCU?

- The value of Free-Run timer 0 can be used as capture data by ICU0 and ICU1
- The value of Free-Run timer 1 can be used as capture data by ICU2 and ICU3
- The value of Free-Run timer 2 can be used as compare data by OCU0 and OCU1
- The value of Free-Run timer 3 can be used as compare data by OCU2 and OCU3
- The value of Free-Run timer 4 can be used as capture data by ICU4 and ICU5
- The value of Free-Run timer 5 can be used as capture data by ICU6 and ICU7
- The value of Free-Run timer 6 can be used as compare data by OCU4 and OCU5
- The value of Free-Run timer 7 can be used as compare data by OCU6 and OCU7
- The value of Free-Run timer 8 can be used as capture data by ICU8 and ICU9

35.9. Caution

- Clearing the Free-Run timer
 - When a reset occurs (the INITX pin input, the watchdog reset, the software reset), the counter is initialized to “0000” and the counting is running.
 - When the Free-Run timer is cleared by software, the counter is cleared and the clear request is generated almost at the same time. If the counter is cleared by the compare-match, it is cleared when it is counted up.
 - After writing “1” in the clear bit (CLR), this request (CLR=“1”) is cleared at the same clear timing of the Free-Run timer. When the clear operation of this CLR and writing “1” to the clear bit occurs at the same time, the clear bit (CLR) keeps “1”, and after the next time the timer is cleared, it is cleared. (As a result, the Free-Run timer is cleared twice.)
 - The counter clear operation (the software, the overflow, and the compare-match) of the Free-Run timer is enabled while the Free-Run timer is counting. To clear while the Free-Run timer is stopped, write 0000_H in the timer count data register.
- Write to the timer data register

When writing the value in the Free-Run timer, make sure to do so while the Free-Run timer is stopped (STOP=“0”), and with word access.
- External clock operation
 - The pulse width required for the external clock is 2/CLKP minimum.
 - When using an external clock, the timing of the compare-match output and the interrupt occurrence is the same as the next count clock timing after the compare-match. Therefore, to allow the compare-match output and interrupt generation, an external clock input of at least 1 clock is required after the compare-match.
- Read/modify/write

The interrupt flag (IVF) is always read as “1” in read/modify/write.
- Interrupt flag

If the interrupt flag set to “1” and cleared to “0” by software simultaneously, the flag setting operation overrides the flag clearing operation.

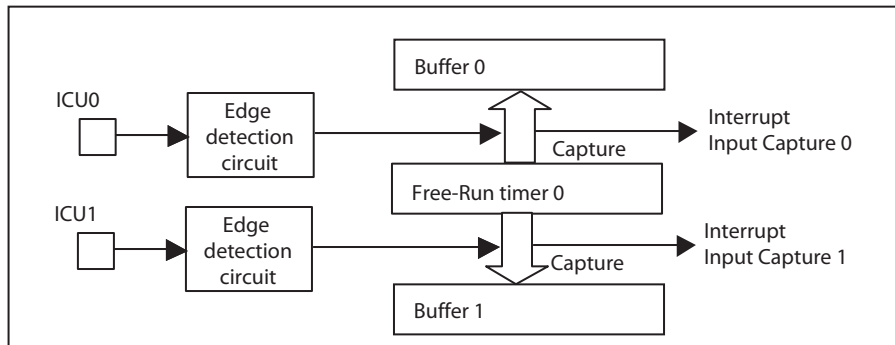
Chapter 36 Input Capture Unit (ICU)

36.1. Overview

The input capture unit (ICU) stores the free-run timer value if an edge is detected at an ICU input pin. Using this stored value it is possible to calculate the time between edges from the external signal.

This unit can be configured to detect rising and/or falling edges.

Figure 36.1-1 Block diagram of Input Capture (Example of channels 0/1)



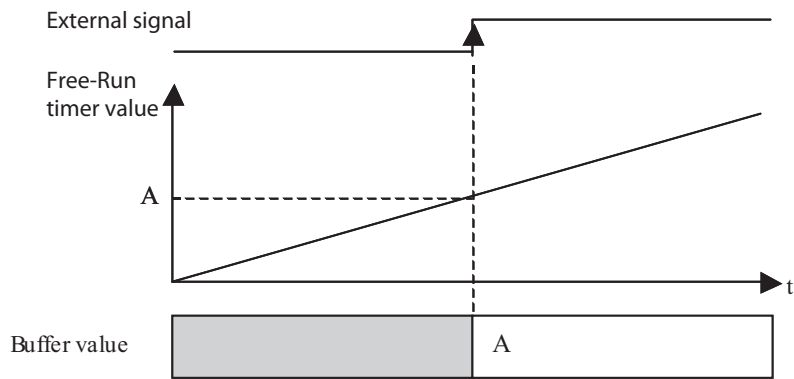
36.2. Features

- Format: Edge detection circuit + 16 bit buffer (capture register)
- Quantity: 4 groups = 8 channels (input capture channels 0/1, 2/3, 4/5, 6/7)
- Quantity: 5 groups = 10 channels (input capture channels 0/1, 2/3, 4/5, 6/7 8/9)

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Compatible timers: Input capture channels 0/1 use Free-Run timer 0
Input capture channels 2/3 use Free-Run timer 1
Input capture channels 4/5 use Free-Run timer 4
Input capture channels 6/7 use Free-Run timer 5
Input capture channels 8/9 use Free-Run timer 8
- Edge Detection: Rising/falling/both edges
- Interrupt: Edge detection
- Capture value: Timer value (0000_H-FFFF_H)
- Timer: Uses Free-Run timer 0
- Precision: 4/CLKP, 16/CLKP, 32/CLKP, 64/CLKP (Free-Run timer count clock)

Figure 36.2-1 Functionality of Input Capture



36.3. Configuration

Figure 36.3-1 Configuration Diagram for Input Capture

Input capture 0-1

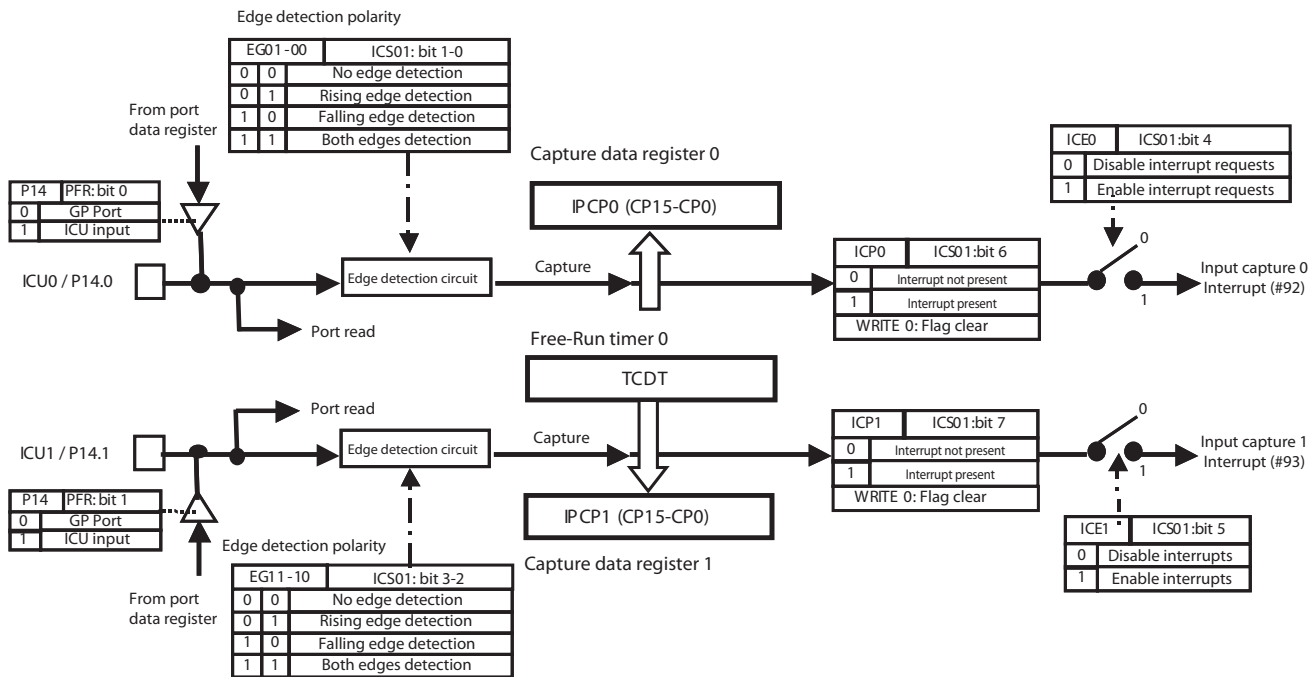


Figure 36.3-2 Register List

Input Capture 0																			
Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000184H		CP15	CP14	CP13	CP12	CP11	CP10	CP9	CP8	CP7	CP6	CP5	CP4	CP3	CP2	CP1	CP0	IPCP0 (Input capture data 0)	
000181H	Bit	7	6	5	4	3	2	1	0										
		ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	ICS01									(Capture control 01)
000402H		ICU7	ICU6	ICU5	ICU4	ICU3	ICU2	ICU1	ICU0	PFR 14								(Port function 14)	
000466H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38								(Interrupt level)	
0FFE8CH		32Bits															(Interrupt vector #92)		

* For information on ICR register and interrupt vector, see the section entitled "Interrupt Control".

Input Capture 1																			
Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000186H		CP15	CP14	CP13	CP12	CP11	CP10	CP9	CP8	CP7	CP6	CP5	CP4	CP3	CP2	CP1	CP0	IPCP1 (Input capture data 1)	
000181H	Bit	7	6	5	4	3	2	1	0										
		ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	ICS01 (Capture control 01)									
000402H		ICU7	ICU6	ICU5	ICU4	ICU3	ICU2	ICU1	ICU0	PFR14 (Port function 14)									
000466H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38 (Interrupt level)									
0FFE88H		32Bits (Interrupt vector #93)																	

* For information on ICR register and interrupt vector, see the section entitled "Interrupt Control".

Note: For information about ICR registers and interrupt vectors, see “Chapter 24 Interrupt Control (Page No.429)”.

36.4. Register

36.4.1 IPCP: Input Capture Data Register

These registers store the Free-Run timer content after detecting an external signal edge (rising and/or falling) on a pin (ICU7 - ICU0).

The IPCP0 and IPCP1 store the content of the Free-Run timer 0.

The IPCP2 and IPCP3 store the content of the Free-Run timer 1.

The IPCP4 and IPCP5 store the content of the Free-Run timer 4.

The IPCP6 and IPCP7 store the content of the Free-Run timer 5.

The IPCP8 and IPCP9 store the content of the Free-Run timer 8.

- **IPCP0 (Input capture 0): Address 0184h (Access: Half-word, Word)**
- **IPCP1 (Input capture 1): Address 0186h (Access: Half-word, Word)**
- **IPCP2 (Input capture 2): Address 0188h (Access: Half-word, Word)**
- **IPCP3 (Input capture 3): Address 018Ah (Access: Half-word, Word)**
- **IPCP4 (Input capture 4): Address 02D4h (Access: Half-word, Word)**
- **IPCP5 (Input capture 5): Address 02D6h (Access: Half-word, Word)**
- **IPCP6 (Input capture 6): Address 02D8h (Access: Half-word, Word)**
- **IPCP7 (Input capture 7): Address 02DAh (Access: Half-word, Word)**
- **IPCP8 (Input capture 8): Address 05F4h (Access: Half-word, Word)**
- **IPCP9 (Input capture 9): Address 05F6h (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
CP15	CP14	CP13	CP12	CP11	CP10	CP9	CP8	
X	X	X	X	X	X	X	X	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	bit
CP7	CP6	CP5	CP4	CP3	CP2	CP1	CP0	
X	X	X	X	X	X	X	X	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

MB91460 Series**36.4.2 ICS: Input Capture Control Register**

- ICS01 (Input capture 0-1): Address 0181h (Access: Byte)
- ICS23 (Input capture 2-3): Address 0183h (Access: Byte)
- ICS45 (Input capture 4-5): Address 02D1h (Access: Byte)
- ICS67 (Input capture 6-7): Address 02D3h (Access: Byte)
- ICS89 (Input capture 8-9): Address 05F1h (Access: Byte)

7	6	5	4	3	2	1	0	bit
ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	
0	0	0	0	0	0	0	0	Initial value
R(RM1),W	R(RM1),W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit7: Input capture 1 interrupt flag

ICP1	Status	
	Read	Write
0	No interrupt present	Clear flag
1	Interrupt present (edge detection present)	No effect on operation

- When the signal change (edge) selected by the active capture edge selection bit (EG[11:10]) is detected on the input from an external pin, the flag is set to “1”.
- To activate the interrupt request it is necessary to set the interrupt request enable bit (ICE1=“1”).
- If the interrupt flag (ICP1) is set to “1” and is cleared to “0” by software simultaneously, the interrupt flag will be set to “1”.

- bit6: Input capture 0 interrupt flag

ICP0	Status	
	Read	Write
0	No interrupt present	Clear flag
1	Interrupt present	No effect on operation

- When the signal change selected by the active capture edge selection bit (EG[01:00]) is detected on the input from an external pin (CS0), the flag is set to “1”.
- To activate the interrupt request, the interrupt request enable setting (ICE1=“1”) is necessary.
- If the interrupt flag (ICP0) is set to “1” and is cleared to “0” by software simultaneously, the interrupt flag will be set to “1”.

- bit5: Input capture 1 interrupt request enable

ICE1	Operation
0	Interrupt request disabled
1	Interrupt request enabled

- If input capture 1 interrupt request enable bit is set to “1”, input capture 1 interrupt request ICP1 will be enabled.

- bit4: Input capture 0 interrupt request enable

ICE0	Operation
0	Interrupt request disabled
1	Interrupt request enabled

- If input capture 0 interrupt request enable bit is set to “1”, input capture 0 interrupt request ICP0 will be enabled.
- bit3-bit2: Input capture 1 active edge selection

EG11	EG10	Edge selection
0	0	Input capture unit is stopped
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising edge and falling edge)

- Select the active capture edge for the input capture signal from external pin (ICU1)
- If the active edge selection bit is “00”, input capture 1 is stopped.
- bit1-bit0: Input capture 0 active edge selection

EG01	EG00	Edge selection
0	0	Input capture unit is stopped
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising edge and falling edge)

- Select the active capture edge for the input capture signal for external pin (ICU0).
- When the active edge selection bit is “00”, input capture 0 is stopped.

MB91460 Series**36.4.3 Addresses and Connections**

ICU8/9 has been added to emulate the corresponding feature of MB91F460Q series.

The following table shows addresses and connections of all ICUs.

Address	ICU	ICU Input is connected to...		ICU Interrupt Nr.	ICU DMA Resource Nr.
		PFR=0 * ¹	PFR=1		
000180 _H to 000188 _H	ICU 0	LIN-USART 0/8 * ²	GP14_0	92	80
	ICU 1	LIN-USART 1/9	GP14_1	93	81
	ICU 2	LIN-USART 2/10	GP14_2	94	82
	ICU 3	LIN-USART 3/11	GP14_3	95	83
0002D0 _H to 0002D8 _H	ICU 4	LIN-USART 4/12	GP14_4	96	84
	ICU 5	LIN-USART 5/13	GP14_5	97	85
	ICU 6	LIN-USART 6/14	GP14_6	98	86
	ICU 7	LIN-USART 7/15	GP14_7	99	87
0005F0 _H to 0005F4 _H	ICU 8	GP22_6	LIN-USART 8 * ³	92 * ⁴	180
	ICU 9	GP22_7	LIN-USART 9 * ⁵	93 * ⁶	181

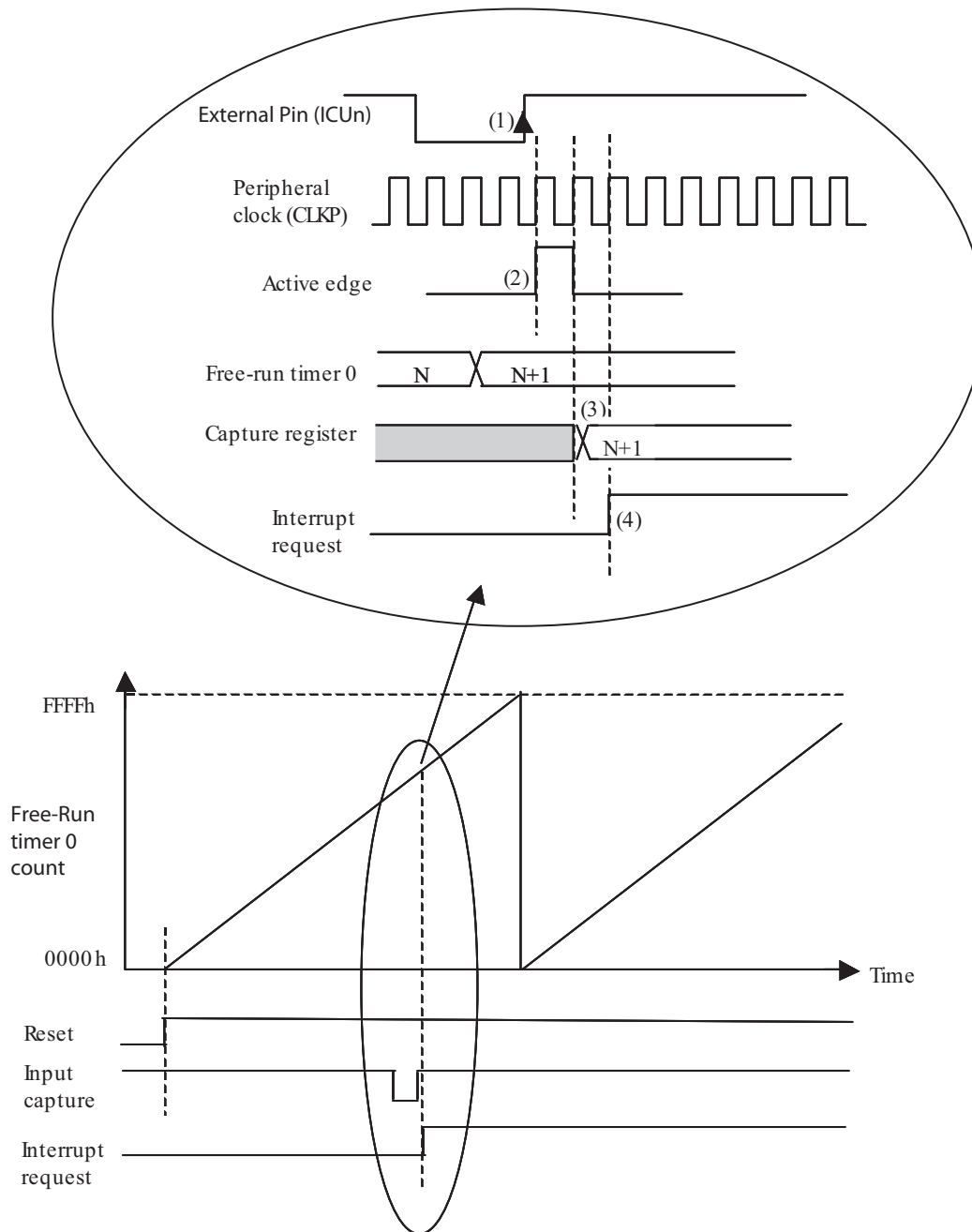
1. PFR is the Port Function Register of the corresponding port.
2. ICU is connected to the LSYN outputs of the LIN-USARTs for LIN break and LIN synchronization field detection. The LSYN lines of the 2 USART modules are connected by OR function.
3. If PFR is set, the port has the I²C resource enabled and ICU is connected to LIN-USART.
4. Interrupt shared with ICU 0
5. If PFR is set, the port has the I²C resource enabled and ICU is connected to LIN-USART.
6. Interrupt shared with ICU 1

36.5. Operation

The input capture operation is described below.

36.5.1 Capture Timing, Interrupt Timing

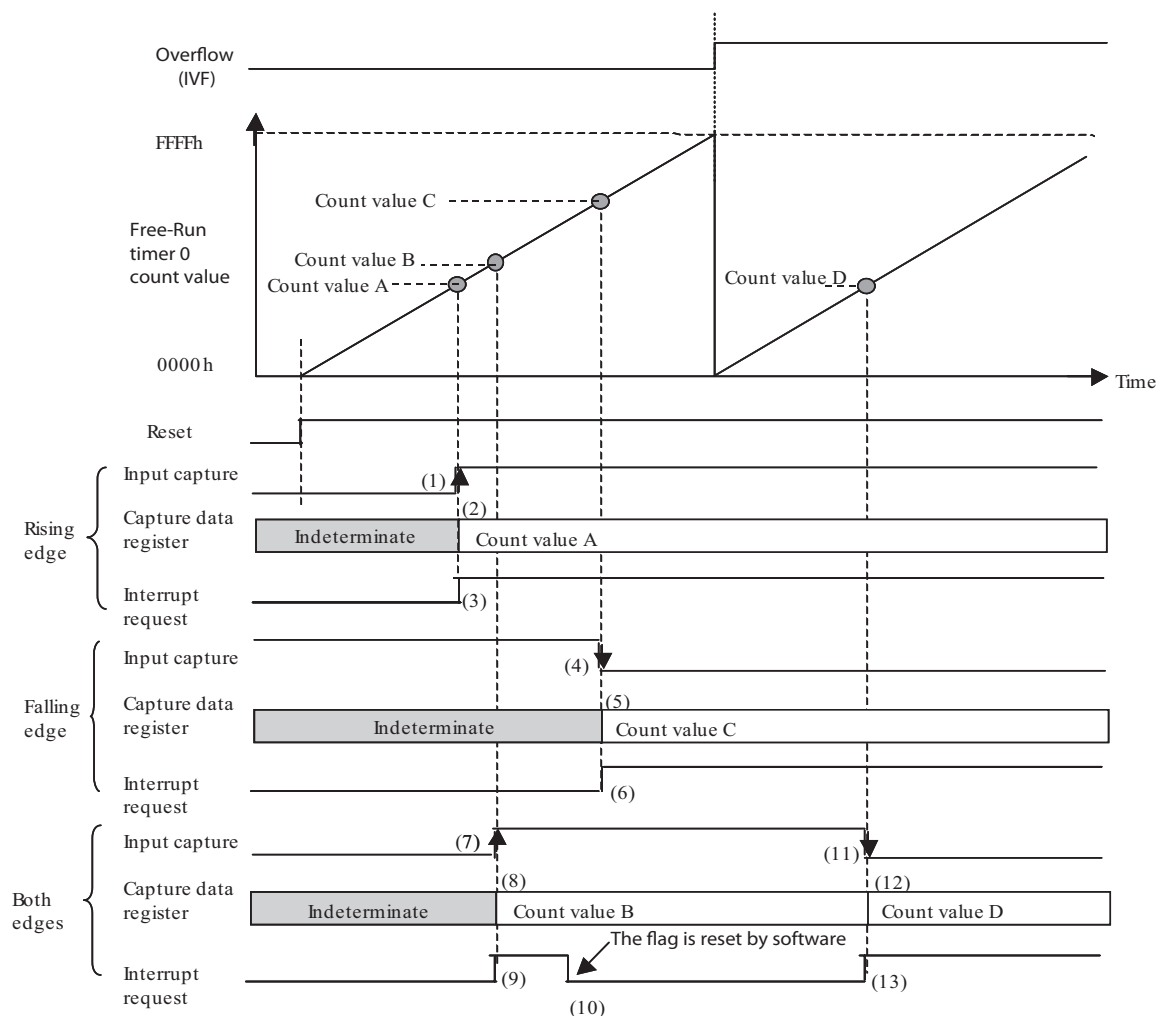
Figure 36.5-1 Operation of Input Capture (Capture Timing, Interrupt Timing)



- (1) Rising edge of input signal
- (2) Internal signal generated by edge detection (synchronous with peripheral clock)
- (3) Store Free-Run timer value in capture register (capture)
- (4) Input capture interrupt generation (ICU0-ICU1="1")

36.5.2 Input Capture Edge Specification and Operation

Figure 36.5-2 Input Capture Edge Specification and Operation



- When specifying rising edge
 - (1) Detection of rising edge of input signal
 - (2) Storage of Free-Run timer value in capture register (capture)
 - (3) Input capture interrupt generation
- When specifying falling edge
 - (4) Detection of input signal falling edge
 - (5) Storage of Free-Run timer value in capture register (capture)
 - (6) Input capture interrupt generation
- Both edges
 - (7) Detection of input signal rising edge
 - (8) Storage of Free-Run timer value in capture register (capture)
 - (9) Input capture interrupt generation
 - (10) Clear interrupt flag (ICS01.ICP0), (ICS01.ICP1) in software
 - (11) Detection of input signal falling edge
 - (12) Storage of Free-Run timer value in capture register (capture)
 - (13) Input capture interrupt generation

36.6. Settings

Table 36.6-1 Settings Necessary for Using Input Capture

Settings	Setting register	Setting procedure*
Free-Run timer settings	See “ Chapter 35 Free-Run Timer (FRT) (Page No.875)”	—
Free-Run timer activation		See 35.4.
Input pin ICU0-ICU9 settings	Port function register (PFR14.0 - PFR14.7) Port function register (PFR22.6, PFR22.7)	See 30.4.4
Active edge polarity selection for external input	Input capture control register (ICS01, ICS23, ICS45, ICS67, ICS89)	See 36.7.1

*: For the setting procedure, refer to the section indicated by the number.

Table 36.6-2 Required Settings for ICU Interrupt

Settings	Settings register	Setting procedure*
Input Capture interrupt vector, Input capture interrupt level settings	See “ Chapter 24 Interrupt Control (Page No.429)”	See 24.4. see 36.7.3
Input capture interrupt settings Interrupt request clear Interrupt request enable	Input capture control register (ICS01, ICS23, ICS45, ICS67, ICS89)	See 36.4.2 see 36.7.5

*: For the setting procedure, refer to the section indicated by the number.

MB91460 Series

36.7. Q&A

36.7.1 What are the types of active edge polarity for external input, and how to select them?

The active edge polarity types consist of rising, falling, and both, for a total of 3, and are set using the external input active edge selection bit (ICS01.EG[01:00]) and (ICS01.EG[11:10]), (ICS23.EG[01:00]) and (ICS23.EG[11:10]), (ICS45.EG[01:00]) and (ICS45.EG[11:10]), (ICS67.EG[01:00]) and (ICS67.EG[11:10]).

Operation	External input active edge polarity bit (EG[01:00]), (EG[11:10])
To select rising edge	Select "01"
To select falling edge	Select "10"
To select both edges	Select "11"

36.7.2 What about setting the external input pins (ICU0-7)?

Use the port function register and extra port function bits (PFR14.x/EPFR14.x).

Operation	Port function (PFR14.x)	Extra Port function (EPFR14.x)
To set it to the external input pins (ICU0)	Set PFR14.0 to "1"	Set EPFR14.0 to "0"
To set it to the external input pins (ICU1)	Set PFR14.1 to "1"	Set EPFR14.1 to "0"
To set it to the external input pins (ICU2)	Set PFR14.2 to "1"	Set EPFR14.2 to "0"
To set it to the external input pins (ICU3)	Set PFR14.3 to "1"	Set EPFR14.3 to "0"
To set it to the external input pins (ICU4)	Set PFR14.4 to "1"	Set EPFR14.4 to "0"
To set it to the external input pins (ICU5)	Set PFR14.5 to "1"	Set EPFR14.5 to "0"
To set it to the external input pins (ICU6)	Set PFR14.6 to "1"	Set EPFR14.6 to "0"
To set it to the external input pins (ICU7)	Set PFR14.7 to "1"	Set EPFR14.7 to "0"
To set it to the external input pins (ICU8)	Set PFR22.6 to "0"	Set EPFR22.6 to "0"
To set it to the external input pins (ICU9)	Set PFR22.7 to "0"	Set EPFR22.7 to "0"

Remark: When setting the Extra port function register EPFR14.x to "1" the corresponding input capture macro is internally connected to the corresponding LIN-USART macro for LIN Sync Field measurement. Hence, with this setting the corresponding ICU channel is not available as external input.

36.7.3 What about interrupt-related registers?

Input capture interrupt vector and input capture interrupt level settings

The relationship among input capture number, interrupt level, and vector is explained in the following table.

For more information on interrupt level and interrupt vectors, see ["Chapter 24 Interrupt Control \(Page No.429\)"](#).

Number	Interrupt vector (Default)	Interrupt level setting bit (ICR[4:0])
Input Capture 0	#92 Address: 0FFE8Ch	Interrupt level register (ICR38) Address: 0466h
Input Capture 1	#93 Address: 0FFE88h	
Input Capture 2	#94 Address: 0FFE84h	Interrupt level register (ICR39) Address: 0467h
Input Capture 3	#95 Address: 0FFE80h	

Input Capture 4	#96 Address: 0FFE7Ch	Interrupt level register (ICR40) Address: 0468h
Input Capture 5	#97 Address: 0FFE78h	
Input Capture 6	#98 Address: 0FFE74h	Interrupt level register (ICR41) Address: 0469h
Input Capture 7	#99 Address: 0FFE70h	
Input Capture 8	#92 Address: 0FFE8Ch	Interrupt level register (ICR38) Address: 0466h
Input Capture 9	#93 Address: 0FFE88h	

Interrupt flags (ICS01.ICP0), (ICS01.ICP1), (ICS23.ICP0), (ICS23.ICP1), (ICS45.ICP0), (ICS45.ICP1), (ICS67.ICP0), (ICS67.ICP1), (ICS89.ICP0), (ICS89.ICP1) are not automatically cleared, so please set the input capture interrupt flag (ICP1, ICP0) to "0" to clear them before returning from interrupt processing.

36.7.4 What are the types of interrupts?

There is only one type of interrupt, and it is generated by input signal edge detection.

36.7.5 How to enable interrupts

Interrupt request enable bit, interrupt flag

Interrupts are enabled via interrupt request enable bit (ICS01.ICE0), (ICS01.ICE1), (ICS23.ICE0), (ICS23.ICE1), (ICS45.ICE0), (ICS45.ICE1), (ICS67.ICE0), (ICS67.ICE1), (ICS89.ICE0), (ICS89.ICE1).

	Interrupt request enable bit (ICE0), (CE1)
Disable interrupt requests	Set to "0"
Enable interrupt requests	Set to "1"

Clearing of interrupts is done using interrupt flag (ICS01.ICP0), (ICS01.ICP1), (ICS23.ICP0), (ICS23.ICP1), (ICS45.ICP0), (ICS45.ICP1), (ICS67.ICP0), (ICS67.ICP1).

	Interrupt flag (ICP0), (ICP1)
Interrupt clear	Write "0"

36.7.6 How to measure the pulse width of the input signal

- "H" Width measurement:

Specify both edges for edge detection.

First detect the rising edge, then detect the falling edge.

Pulse width = {value stored during falling (input capture register value)
+ "10000h" x Overflow frequency
– value stored during rising (input capture register value)}
x Count clock width of Free-Run timer

Example: value stored during falling = 2320h, Value stored during rising = A635h,
Overflow frequency = 1, count clock = 125ns

=> pulse width = (2320h+10000h-A635h) x 125ns = 3997.375us

- Cycle measurement:

Specify rising (or falling) for edge detection.

Detect edge 2 times.

Cycle = {Second stored value (input capture register value)
+ "10000h" x Overflow frequency
– First stored value (input capture register value)}
x Count clock width of Free-Run timer

36.8. Caution

- Input capture data register
The value of the input capture data register during reset is indeterminate.
Read out of the input capture data register must always be done using 16 or 32 bit access.
- Read modify write
Input capture interrupt flag (ICP0), (ICP1) will be read as “1” when read with read modify write.

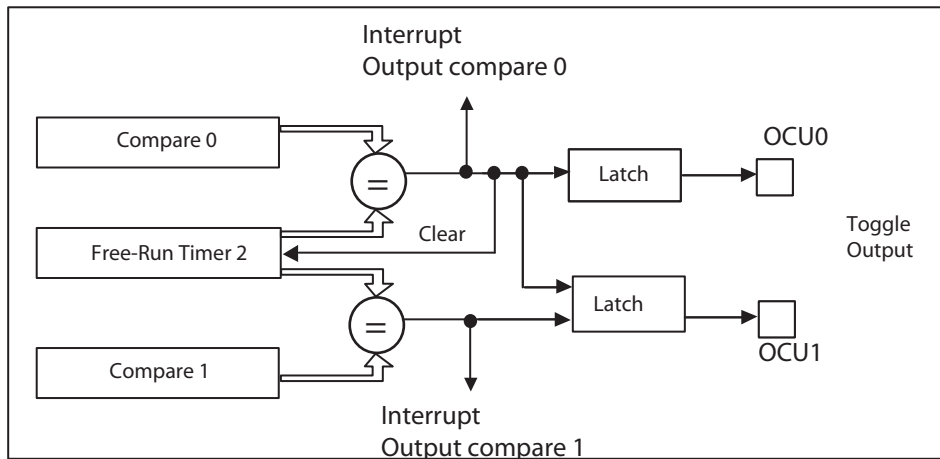
MB91460 Series

Chapter 37 Output Compare Unit (OCU)

37.1. Overview

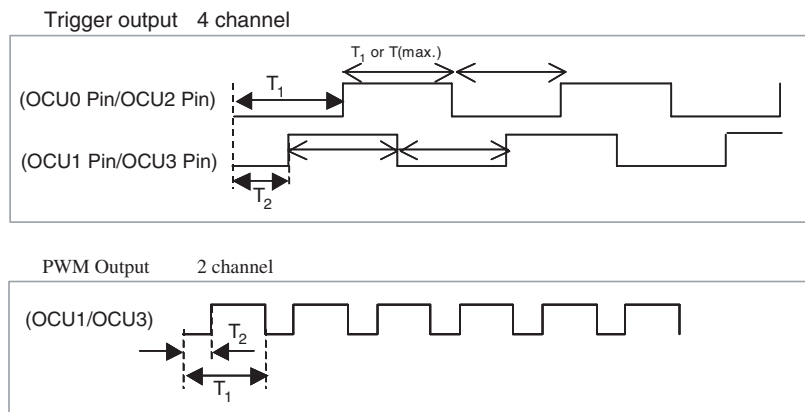
The Output compare unit reverses the level of the output pin (OCUn), if the compare register value matches the timer value of the Free-Run timer.

Figure 37.1-1 Structure of output compare (Example of channels 0/1)



37.2. Features

Figure 37.2-1 Output waveform: Toggle output 4 channel



- Format: 16 bit compare register + Compare circuit
- Quantity: 4 groups = 8 channels (output compare channels 0/1, 2/3, 4/5, 6/7)

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Compatible timers: Output compare channels 0/1 use Free-Run timer 2
Output compare channels 2/3 use Free-Run timer 3
Output compare channels 4/5 use Free-Run timer 6
Output compare channels 6/7 use Free-Run timer 7
- Operation on compare match:
 - Reversal of pin output value (toggle output)

- Free-Run timer clear
- Interrupt generation
- Count precision: 4/CLKP, 16/CLKP, 32/ CLKP, 64/ CLKP (dependent on Free-Run timer)
- Toggle change width (T): 1 x count precision - 10000_H x count precision
- Interrupt: Compare-match interrupt
- Others: Setting of initial output level value is possible (“H”/“L”)

Pins not used for OCU output can be used as general-purpose ports

37.3. Configuration Diagram

Figure 37.3-1 Configuration Diagram for Output compare

Output Compare 0-1

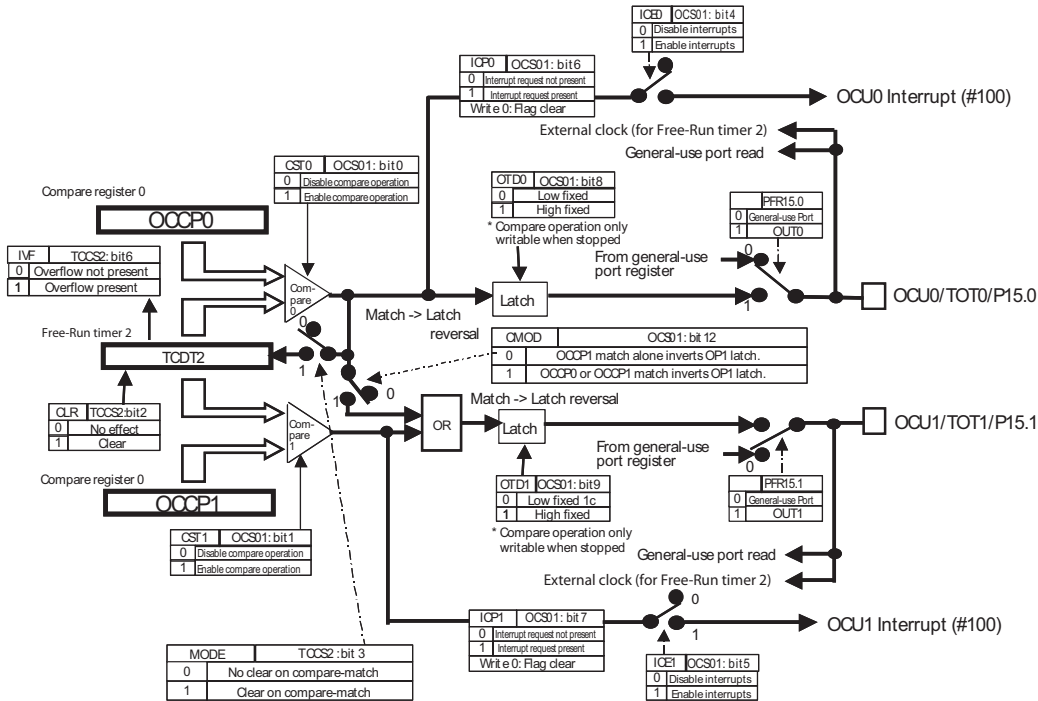


Figure 37.3-2 Register List

Output Compare 0-1																		
Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000190H		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	OCCP0 (Compare 0)
000191H		C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	OCCP1 (Compare 1)
00018CH		---	---	---	CMOD	---	---	OTD1	OTD0	ICP1	ICP0	ICE1	ICE0	---	---	---	---	OCS01 (Output compare control 01)
0001F8H		T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0	TCDT2 (Free-run timer 2)
0001FBH	Bit	7	6	5	4	3	2	1	0									
		ECLK	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0	TCCS2 (Timer control 2)								
000D8FH		OCU7	OCU6	OCU5	OCU4	OCU3	OCU2	OCU1	OCU0	PFR15 (Port function 15)								
00046AH		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR42 (Interrupt level #100/#101)								
0FFE6CH																		(Interrupt vector #100)
0FFE68H																		(Interrupt vector #101)

* For information about ICR register and interrupt vectors, see the section entitled "Interrupt Control".

Note: For information about ICR registers and interrupt vectors, see “Chapter 24 Interrupt Control (Page No.429)”.

MB91460 Series**37.4. Registers****37.4.1 OCS: Output Control Register**

A register for controlling the operation of output compare.

- **OCS01 (Output compare 0-1): Address 018Ch (Access: Byte, Half-word, Word)**
- **OCS23 (Output compare 2-3): Address 018Eh (Access: Byte, Half-word, Word)**
- **OCS45 (Output compare 4-5): Address 02DCh (Access: Byte, Half-word, Word)**
- **OCS67 (Output compare 6-7): Address 02DEh (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
–	–	–	CMOD	–	–	OTD1	OTD0	
1	1	1	0	1	1	0	0	Initial Value
R1/W1	R1/W1	R1/W1	R/W	R1/W1	R1/W1	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
ICP1	ICP0	ICE1	ICE0	–	–	CST1	CST0	
0	0	0	0	1	1	0	0	Initial Value
R(RM1),W	R(RM1),W	R/W	R/W	R1/W1	R1/W1	R/W	R/W	Attribute

(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit15-bit13: undefined Writing does not affect the operation. The read out value is “1”.
- bit12: Reverse Mode

CMOD	Operation Mode
0	Independent operation (the output level reversal operation of pins OCU0-OCU1 is independent)
1	Combined operation (OCU1 output pin level is inverted when output compare 0 or output compare 1 is matched in the compare operation.)

- Specifies the output level reversal operation of pin OCU1 when Free-Run timer count value TCDT2 matches compare registers OCCP0, OCCP1.
- When the reverse mode bit is set to “1”, the operation is as follows:
OCU0 pin: output reverses when Free-Run timer TCDT2 matches compare register 0 (OCCP0)
OCU1 pin: output reverses when Free-Run timer TCDT2 matches compare register 1 (OCCP1)
- When the reverse mode bit is set to “0”, the operation is as follows:
OCU0 pin: output reverses when Free-Run timer TCDT2 matches compare register 0(OCCP0)
OCU1 pin: output reverses when Free-Run timer TCDT2 matches compare register 0 (OCCP0) or compare register 1 (OCCP1)

Note: Reversal mode does not allow interrupts, even with cooperative operation (CMOD=“1”).

- For output from pins OCU0, OCU1, registers PFR15.0, PFR15.1 must be set.
- bit11-bit10: Undefined Writing does not affect the operation. The read value is “1”.
- bit9: Pin-level settings (output compare 1)

OTD1	Operation
0	Set the output level of pin OCU1 to “L”
1	Set the output level of pin OCU1 to “H”

To perform output on pin OCU1, general-purpose port settings must be performed.

- bit8: Pin-level settings (output compare 0)

OTD0	Operation
0	Set the output level of pin OCU0 to “L”
1	Set the output level of pin OCU0 to “H”

- To perform output on pin OCU0, general-purpose port settings must be performed.

- bit7: Interrupt flag (output compare 1)

ICP1	Status	
	Read	Write
0	No Interrupt present	Clear flag (ICP1)
1	Interrupt present	No effect on operation

- If Free-Run timer count value TCDT2 matches the output compare register OCCP1, ICP1 becomes “1”.
- Interrupt request is enabled when the interrupt enable bit (ICP1) is set to “1”.
- If the interrupt request and the flag resetting occur simultaneously, the flag is set to “1” (The flag setting has higher priority than clearing the flag).
- When using an external clock as the Free-Run timer operation clock, at least one external clock input is necessary after compare match for output compare-match output and interrupt generation.
- bit6: Interrupt flag (output compare 0)

ICP0	Status	
	Read	Write
0	No Interrupt present	Clear flag (ICP0)
1	Interrupt present	No effect on operation

- If Free-Run timer count value TCDT2 matches output compare register OCCP0, ICP0 becomes “1”.
- Interrupt request is enabled when the interrupt enable bit (ICP0) is “1”.
- If the interrupt request and the flag resetting occur simultaneously, the flag is set to “1” (The flag setting has higher priority than clearing the flag).
- When using an external clock as the Free-Run timer operation clock, at least one external clock input is necessary after compare match for output compare-match output and interrupt generation.
- bit5: Interrupt request enable (output compare 1)

ICE1	Status
0	Disable output compare 1 interrupt requests
1	Enable output compare 1 interrupt requests

- bit4: Interrupt request enable (output compare 0)

ICE0	Status
0	Disable output compare 0 interrupt requests
1	Enable output compare 0 interrupt requests

- bit3-bit2: Undefined Writing does not affect the operation. The read value is always “1”.
- bit1: Enable operation requests (output compare 1)

CST1	Operation
0	Stop operation of output compare 1
1	Enable operation of output compare 1

- A bit that enables a comparison operation between the Free-Run timer count value and the output compare register (TCDT2 and OCCP1).
- Before enabling the operation, always set a value to compare register OCCP1.
- If the Free-Run timer is stopped, output compare also stops.

- bit0: Enable operation requests (output compare 0)

CST0	Operation
0	Disable output compare 0 operation
1	Enable output compare 0 operation

- A bit that enables a comparison operation between the Free-Run timer count value and the output compare register (TCDT2 and OCCP0).
- Before enabling the operation, always set a value to compare register OCCP0.
- If the Free-Run timer is stopped, output compare also stops.

MB91460 Series**37.4.2 OCCP: Compare Register**

Register that sets the value to be compared to the 16 bit Free-Run timer count value.

- **OCCP0 (OCU0): Address 0190h (Access: Half-word, Word)**
- **OCCP1 (OCU1): Address 0192h (Access: Half-word, Word)**
- **OCCP2 (OCU2): Address 0194h (Access: Half-word, Word)**
- **OCCP3 (OCU3): Address 0196h (Access: Half-word, Word)**
- **OCCP4 (OCU4): Address 02E0h (Access: Half-word, Word)**
- **OCCP5 (OCU5): Address 02E2h (Access: Half-word, Word)**
- **OCCP6 (OCU6): Address 02E4h (Access: Half-word, Word)**
- **OCCP7 (OCU7): Address 02E6h (Access: Half-word, Word)**

15	14	13	12	11	10	9	8	bit
C15	C14	C13	C12	C11	C10	C9	C8	
X	X	X	X	X	X	X	X	Initial Value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
C7	C6	C5	C4	C3	C2	C1	C0	
X	X	X	X	X	X	X	X	Initial Value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

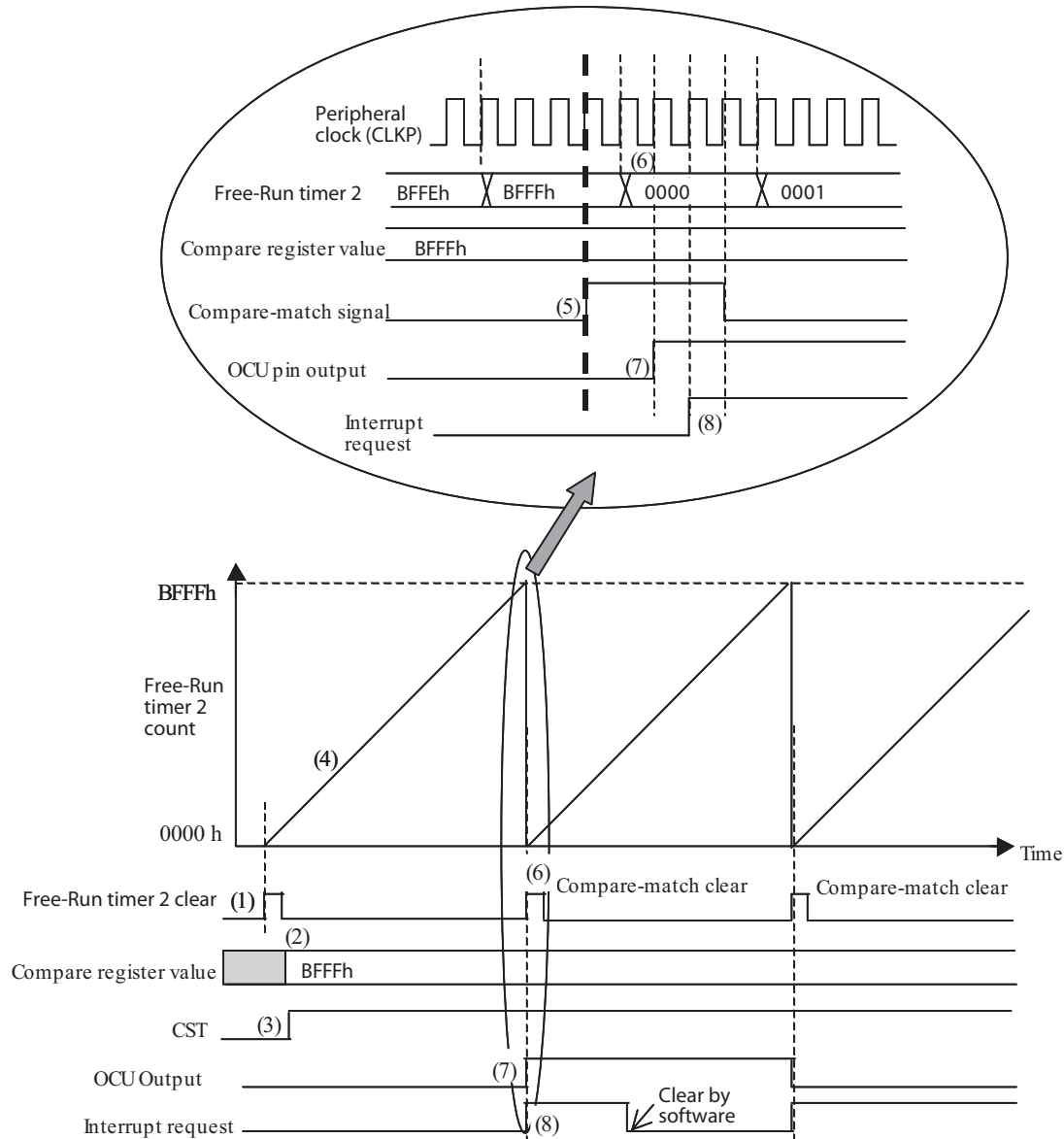
(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- The values of the OCCP0 and OCCP1 compare registers are compared with the TCD2 value of the Free-Run timer².
- The values of the OCCP2 and OCCP3 compare registers are compared with the TCD3 value of the Free-Run timer³.
- The values of the OCCP4 and OCCP5 compare registers are compared with the TCD6 value of the Free-Run timer⁶.
- The values of the OCCP6 and OCCP7 compare registers are compared with the TCD7 value of the Free-Run timer⁷.

37.5. Operation

37.5.1 Output Compare Output (Independent Reversal) CMOD="0"

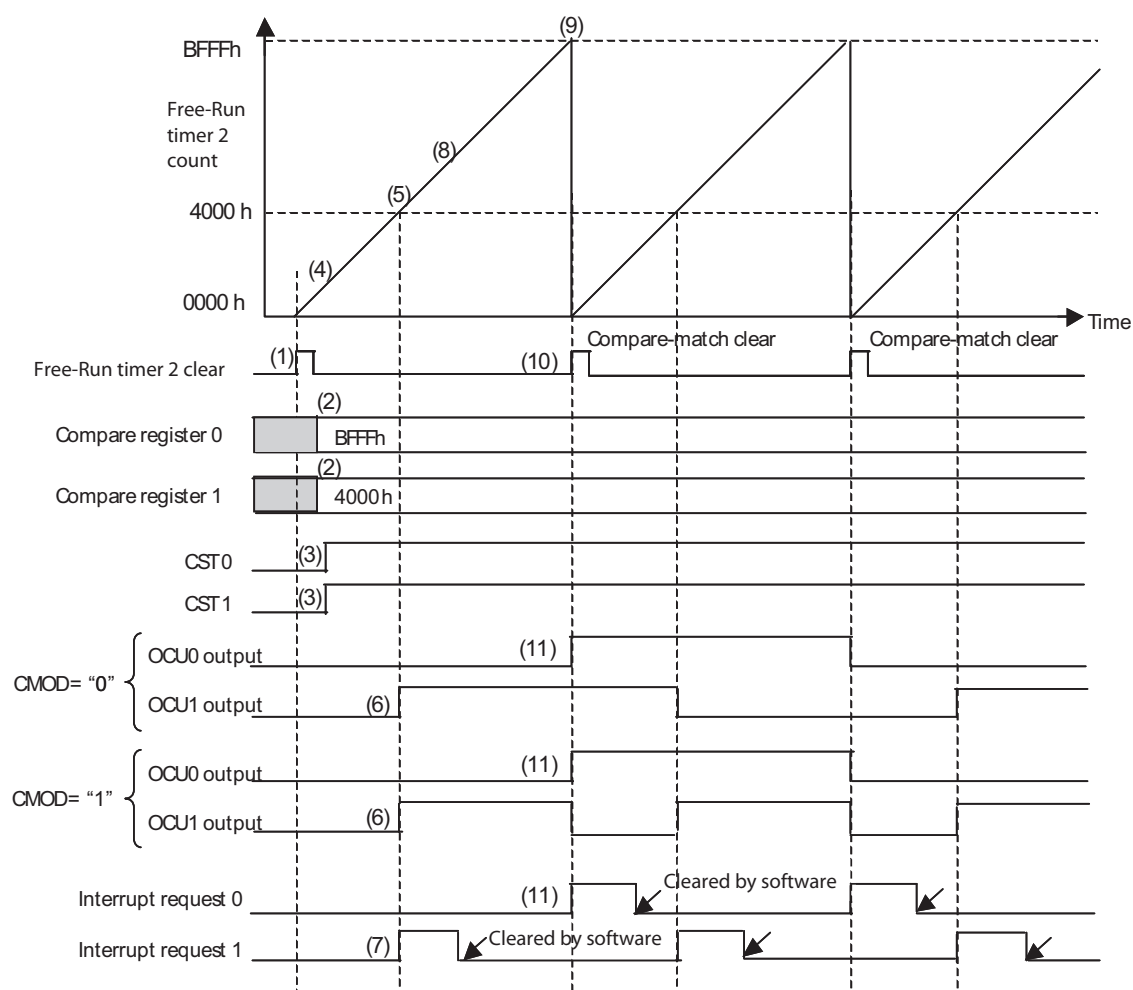
Figure 37.5-1 Output Compare Output (CMOD="0")



- (1) Free-Run timer clear/reset
- (2) Compare value setting
- (3) Enable compare operation (CST="1")
- (4) Free-Run timer count up (example of 1 clock in 4)
- (5) Compare Free-Run timer value and compare value and match (compare match).
- (6) Free-Run timer clear from compare match (Free-Run timer 2)
- (7) OCU output level reversal
- (8) Compare match interrupt request generation

37.5.2 Output Compare Output (Cooperative Reversal) CMOD="1"

Figure 37-5-2 Output Compare Output (CMOD="1")



- (1) Free-Run timers clear/reset (only through Compare 0, not through Compare 1)
- (2) Compare 0 and compare 1 value settings
- (3) Enable compare operation
- (4) Free-Run timer count up
- (5) Compare 1 match
- (6) OCU1 output level reversal
- (7) Compare 1 match interrupt
- (8) Free-Run timer count up
- (9) Compare 0 match
Free-Run timer is cleared (just if Compare 0 match occurs, not for compare 1 match)
- (10) OCU0 output level reversal
When CMOD="1", OCU1 output level also reverses
- (11) Compare 0 match interrupt

37.6. Settings

Table 37.6-1 Settings Necessary for Using Output Compare

Settings	Setting Register	Setting Procedure*
Free-Run timer setting	See “ Chapter 35 Free-Run Timer (FRT) (Page No.875)”	See 35.4 .
Compare value setting	Compare register (OCCP0 - OCCP7)	See 37.7.1
Compare mode setting	Output control register (OCS01, OCS23, OCS45, OCS67)	See 37.7.2
Stop compare operation		See 37.7.3
Set initial level of compare pin output		See 37.7.4
Set OCU0-OCU7 pins to output	Port function register (PFR15.0 - PFR15.7) Extra port function register (EPFR15.0 - EPFR15.7)	See 37.7.5
Clear Free-Run timer	Timer control register (TCCS2, TCCS3, TCCS6, TCCS7) See “ Chapter 35 Free-Run Timer (FRT) (Page No.875)”	See 37.7.6
Enable compare operation (activate)	Output control register (OCS01, OCS23, OCS45, OCS67)	See 37.7.7

*: For the setting procedure, refer to the section indicated by the number.

Table 37.6-2 Item Necessary to Clear the Free-Run Timer upon Compare-match.

Setting	Setting Register	Setting Procedure*
Select Free-Run timer clear mode	Timer control register (TCCS2, TCCS3, TCCS6, TCCS7) See “ Chapter 35 Free-Run Timer (FRT) (Page No.875)”	See 37.7.8

*: For the setting procedure, refer to the section indicated by the number.

Table 37.6-3 Item Necessary for Performing Interrupts

Setting	Setting register	Setting Procedure*
Output compare interrupt vector, output compare interrupt level setting	See “ Chapter 24 Interrupt Control (Page No.429)”	See 37.7.9
Output compare interrupt setting Clear interrupts Enable interrupt requests	Output control register (OCS01, OCS23, OCS45, OCS67)	See 37.7.11

*: For the setting procedure, refer to the section indicated by the number.

37.7. Q & A

37.7.1 How to set the compare value

Write the compare value to compare registers OCCP0 - OCCP7.

37.7.2 How to set the compare mode (for OCU1, OCU3, OCU5, OCU7 output)

It is set using the corresponding compare mode bits (OCS01.CMOD), (OCS23.CMOD), (OCS45.CMOD), (OCS67.CMOD).

Operation	Compare mode bit
To reverse OCU1 output using a compare-match from only Free-Run timer 2 and compare register 1	Set OCS01.CMOD bit to "0"
To reverse OCU3 output using a compare-match from only Free-Run timer 3 and compare register 3	Set OCS23.CMOD bit to "0"
To reverse OCU5 output using a compare-match from only Free-Run timer 6 and compare register 5	Set OCS45.CMOD bit to "0"
To reverse OCU7 output using a compare-match from only Free-Run timer 7 and compare register 7	Set OCS67.CMOD bit to "0"
To reverse OCU1 output using a compare-match from Free-Run timer 2 and compare register 0, as well as Free-Run timer 0 and compare register 1	Set OCS01.CMOD bit to "1"
To reverse OCU3 output using a compare-match from Free-Run timer 3 and compare register 2, as well as Free-Run timer 1 and compare register 3	Set OCS23.CMOD bit to "1"
To reverse OCU5 output using a compare-match from Free-Run timer 6 and compare register 4, as well as Free-Run timer 4 and compare register 1	Set OCS45.CMOD bit to "1"
To reverse OCU7 output using a compare-match from Free-Run timer 7 and compare register 6, as well as Free-Run timer 5 and compare register 3	Set OCS67.CMOD bit to "1"

With no relation to CMOD bit,

OCU0 output is reversed by a compare-match between Free-Run timer 2 and compare register 0 only.

OCU2 output is reversed by a compare-match between Free-Run timer 3 and compare register 2 only.

OCU4 output is reversed by a compare-match between Free-Run timer 6 and compare register 4 only.

OCU6 output is reversed by a compare-match between Free-Run timer 7 and compare register 6 only.

37.7.3 How to enable/disable the compare operation

It is enabled/disabled via the compare operation enable bit (OCS01.CST[1:0]), (OCS23.CST[1:0]), (OCS45.CST[1:0]), (OCS67.CST[1:0]).

Operation	Compare	Compare operation permission bit
To stop (disable) the compare operation	Compare 0	Set OCS01.CST[0] to "0"
	Compare 1	Set OCS01.CST[1] to "0"
	Compare 2	Set OCS23.CST[0] to "0"
	Compare 3	Set OCS23.CST[1] to "0"
	Compare 4	Set OCS45.CST[0] to "0"
	Compare 5	Set OCS45.CST[1] to "0"
	Compare 6	Set OCS67.CST[0] to "0"
	Compare 7	Set OCS67.CST[1] to "0"

To enable compare operation	Compare 0	Set OCS01.CST[0] to "1"
	Compare 1	Set OCS01.CST[1] to "1"
	Compare 2	Set OCS23.CST[0] to "1"
	Compare 3	Set OCS23.CST[1] to "1"
	Compare 4	Set OCS45.CST[0] to "1"
	Compare 5	Set OCS45.CST[1] to "1"
	Compare 6	Set OCS67.CST[0] to "1"
	Compare 7	Set OCS67.CST[1] to "1"

37.7.4 How to set the initial level of the compare pin output

Set it with compare pin output specification bit (OCS01.OTD[1:0]), (OCS23.OTD[1:0]), (OCS45.OTD[1:0]), (OCS67.OTD[1:0]).

Operation	Compare pin output specification bit
To set compare 0 pin to "L"	Set OCS01.OTD0 to "0"
To set compare 0 pin to "H"	Set OCS01.OTD0 to "1"
To set compare 1 pin to "L"	Set OCS01.OTD1 to "0"
To set compare 1 pin to "H"	Set OCS01.OTD1 to "1"
To set compare 2 pin to "L"	Set OCS23.OTD0 to "0"
To set compare 2 pin to "H"	Set OCS23.OTD0 to "1"
To set compare 3 pin to "L"	Set OCS23.OTD1 to "0"
To set compare 3 pin to "H"	Set OCS23.OTD1 to "1"
To set compare 4 pin to "L"	Set OCS45.OTD0 to "0"
To set compare 4 pin to "H"	Set OCS45.OTD0 to "1"
To set compare 5 pin to "L"	Set OCS45.OTD1 to "0"
To set compare 5 pin to "H"	Set OCS45.OTD1 to "1"
To set compare 6 pin to "L"	Set OCS67.OTD0 to "0"
To set compare 6 pin to "H"	Set OCS67.OTD0 to "1"
To set compare 7 pin to "L"	Set OCS67.OTD1 to "0"
To set compare 7 pin to "H"	Set OCS67.OTD1 to "1"

37.7.5 How to set the output for compare pins OCU0-OCU7

The pin is configured as output from the OCU through the PFR15[7:0]

Operation	Port function bit	Extra port function bit
To set compare 0 pin (OCU0) to output	Set PFR15.0 bit to "1"	Set EPFR15.0 bit to "0"
To set compare 1 pin (OCU1) to output	Set PFR15.1 bit to "1"	Set EPFR15.1 bit to "0"
To set compare 2 pin (OCU2) to output	Set PFR15.2 bit to "1"	Set EPFR15.2 bit to "0"
To set compare 3 pin (OCU3) to output	Set PFR15.3 bit to "1"	Set EPFR15.3 bit to "0"
To set compare 4 pin (OCU4) to output	Set PFR15.4 bit to "1"	Set EPFR15.4 bit to "0"
To set compare 5 pin (OCU5) to output	Set PFR15.5 bit to "1"	Set EPFR15.5 bit to "0"
To set compare 6 pin (OCU6) to output	Set PFR15.6 bit to "1"	Set EPFR15.6 bit to "0"
To set compare 7 pin (OCU7) to output	Set PFR15.7 bit to "1"	Set EPFR15.7 bit to "0"

37.7.6 How to clear the Free-Run timer

The Free-Run timer is cleared through clear bits (TCCS2.CLR), (TCCS3.CLR), (TCCS6.CLR), (TCCS7.CLR).

Operation	Clear Bit (CLR)
To clear the Free-Run timer	Write "1"

For other methods, see "[Chapter 35 Free-Run Timer \(FRT\) \(Page No.875\)](#)".

37.7.7 How to enable the compare operation

Enable it through compare operation enable bit (OCS01.CST[1:0]), (OCS23.CST[1:0]), (OCS45.CST[1:0]), (OCS67.CST[1:0]).

See "[37.7.4 How to set the initial level of the compare pin output \(Page No.914\)](#)".

37.7.8 How to compare the Free-Run timer value with the compare register value and clear the Free-Run timer when they match

It is compared and cleared through timer initialization condition bit (TCCS2.MODE), (TCCS3.MODE), (TCCS6.MODE), (TCCS7.MODE).

Operation	Timer initialization condition bit (MODE)
To clear Free-Run timer upon compare 0 match	Set TCCS2.MODE to "1"
To clear Free-Run timer upon compare 2 match	Set TCCS3.MODE to "1"
To clear Free-Run timer upon compare 4 match	Set TCCS6.MODE to "1"
To clear Free-Run timer upon compare 6 match	Set TCCS7.MODE to "1"

37.7.9 What are the interrupt-related registers?

Set the output compare interrupt vector and output compare interrupt level.

The relationship between output compare number, interrupt level, and vector is shown in the following table.

For detailed information on interrupt levels and interrupt vectors, see “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

Number	Interrupt vector (default)	Interrupt level setting bit (ICR[4:0])
Output Compare 0	#100 Address: 0FFE6Ch	Interrupt level register (ICR42) Address: 046Ah
Output Compare 1	#101 Address: 0FFE68h	
Output Compare 2	#102 Address: 0FFE64h	Interrupt level register (ICR43) Address: 046Bh
Output Compare 3	#103 Address: 0FFE60h	
Output Compare 4	#104 Address: 0FFE5Ch	Interrupt level register (ICR44) Address: 046Ch
Output Compare 5	#105 Address: 0FFE58h	
Output Compare 6	#106 Address: 0FFE54h	Interrupt level register (ICR45) Address: 046Dh
Output Compare 7	#107 Address: 0FFE50h	

Interrupt flags (OCS01. ICP[1:0]), (OCS23. ICP[1:0]), (OCS45. ICP[1:0]), (OCS67. ICP[1:0]), are not automatically cleared, so write “0” to the ICP[7:0] bit before returning from interrupt processing to clear them.

37.7.10 What are the types of interrupts?

There is only one type of interrupt, generated upon a compare-match.

37.7.11 How to enable interrupts

Enabling of interrupts is done with interrupt request enable bit (OCS01. ICE[1:0]), (OCS23. ICE[1:0]), (OCS45. ICE[1:0]), (OCS67. ICE[1:0]).

	Interrupt request enable bit (ICE0, ICE1)
Interrupt request disabled	Set to "0"
Interrupt request enabled	Set to "1"

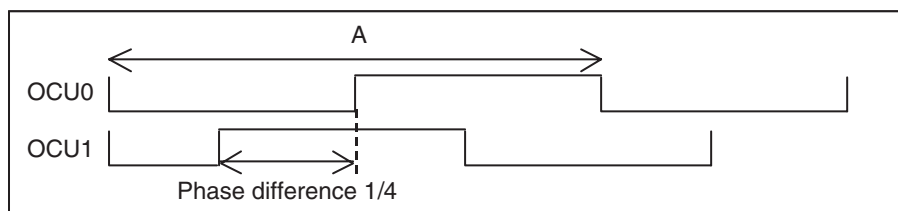
Interrupts are cleared with interrupt flags (OCS01.ICP[1:0]), (OCS23.ICP[1:0]), (OCS45.ICP[1:0]), (OCS67.ICP[1:0]).

	Interrupt flag (ICP0, ICP1)
Interrupt clear	Write "0"

37.7.12 Compare value calculation procedure

- Toggle output pulse

Figure 37.7-1 (Example) To output a period: A, phase difference 1/4 phase pulse



Formula: Compare 0 value = $(A/2) / \text{count clock}$
 Compare 1 value = $(A/4) / \text{count clock}$
 (Count clock: time set with Free-Run timer)

Note: To clear Free-Run timer 2 on compare 0 match setting (TCCS2.MODE="1") and CMOD="0" setting are necessary.

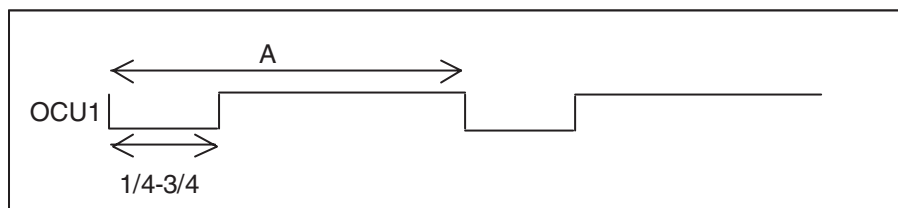
Calculation example: $A=1024\mu\text{s}$, count clock = 125ns

Compare 0 value = $(1024000 / 2) / 125 - 1 = 4095 = \text{FFFh}$

Compare 1 value = $(1024000 / 4) / 125 - 1 = 1023 = \text{7FFh}$

- PWM output

Figure 37.7-2 (Example) To output a period: A, duty 1/4 - 3/4 ("L") PWM,



Formula: Compare 0 value = $A / \text{count clock}$
 Compare 1 value = $(A/4) / \text{count clock}$ (when duty 1/4)
 $(A \times 3/4) / \text{count clock}$ (when duty 3/4)
 (count clock: time set with Free-Run timer)

Note: To clear Free-Run timer 0 on compare 0 match setting (TCCS0.MODE="1") and CMOD="1" setting are necessary.

Calculation example: $A=1024\mu\text{s}$, count clock = 125ns

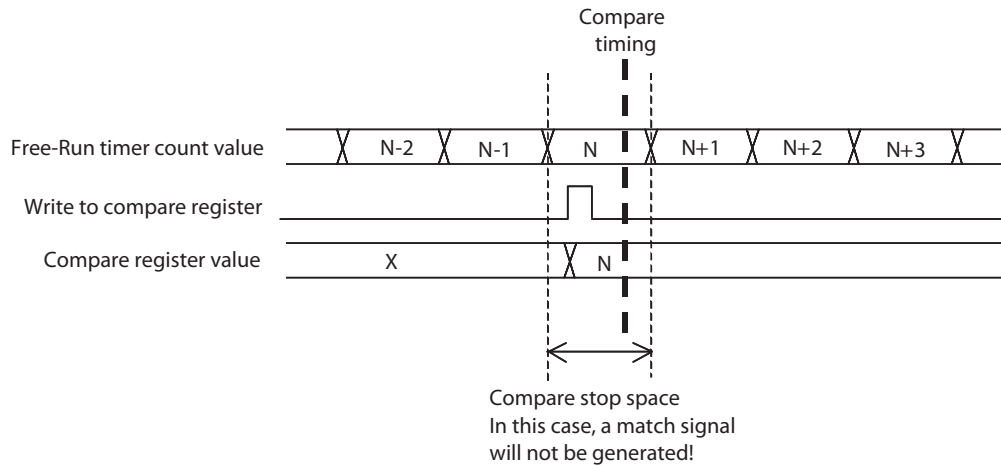
Compare 0 value = $1024000 / 125 - 1 = 8191 = \text{1FFFh}$

Compare 1 value = $(1024000 / 4) / 125 - 1 = 1023 = 7FFh$ (when duty 1/4)
 $(1024000 \times 3 / 4) / 125 - 1 = 1023 = BFFh$ (when duty 3/4)

37.8. Caution

- Compare stop space during compare operation
As shown below, for one count directly after the compare value is written to the compare register, the compare operation cannot be used.

Figure 37.8-1 Compare stop space



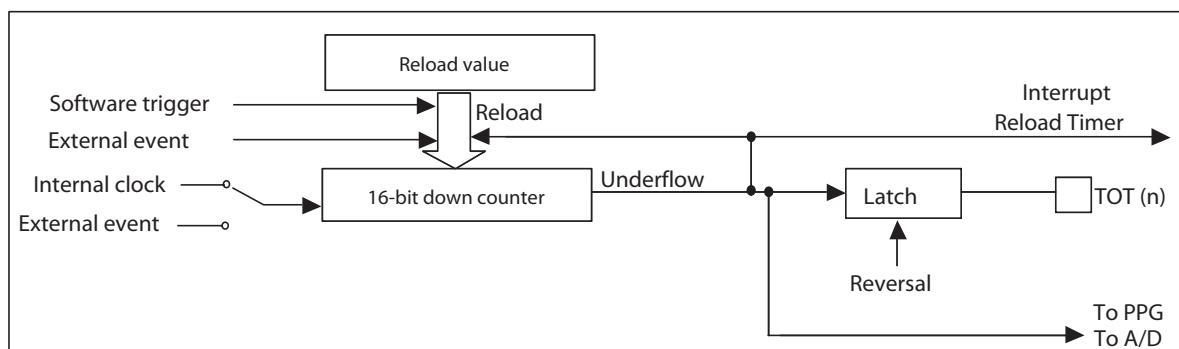
- When CMOD="1" and OCCP0=OCCP1 setting, if a compare match is generated, the port will only reverse once.
- Compare registers (OCCP0 - OCCP7) are set to the initial values. Always set a value before activating them.
- The output compare operation needs to be disabled during the configuration of the output level.
- Output compare is synchronous with the Free-Run timer, so if the Free-Run timer is stopped the compare operation also stops.
- Even when reversal mode specification (CMOD) is set to "1" and the compare operation is in cooperative mode, interrupts are generated independently.
- When using an external clock as the Free-Run timer, compare-matches and interrupts are generated with the following clock edge. To generate compare match output and interrupts, at least 1 clock cycle from prescaled clock must be input to the external clock pin from the Free-Run timer after the compare-match.

Chapter 38 Reload Timer (RLT)

38.1. Overview

The reload timer basically consists of a 16 bit down counter. This counter is decremented either by an external trigger or by software. There is an external output pin associated to this module whose polarity level changes if an underflow condition occurs.

Figure 38.1-1 Block diagram of the Reload Timer



38.2. Features

Figure 38.2-1 Operation: 2 types of operation are possible

- One-shot Operation

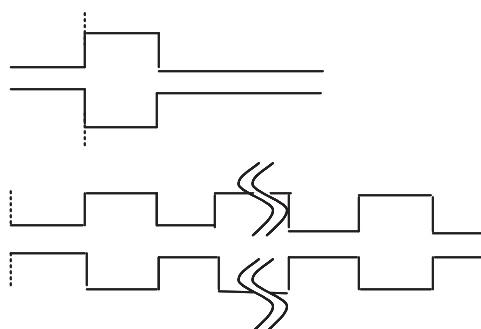
Initial output level

Reversed output level

- Reload Operation

Initial output level

Reversed output level



Format: 16 bit down counter with reload register

Quantity: up to 16 (Output: 8 channels TOT[0-7])

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

Clock mode: Select from two modes

- Internal clock mode

Count clock: CLKP/2, CLKP/8, CLKP/32, CLKP/64, CLKP/128

Activation triggers (4 types)

- External event clock mode

Count clock : External event (TIN[15:0] pins)

Count active edge : Rising/falling/both edges of external event

Activation trigger : Software trigger

Cycle : Cycle = count clock x (reload value + 1)

(Example) When count clock = 16MHz, reload value = 15999

Cycle = $62.5\text{ns} \times (15999+1) = 1.0\text{ms}$

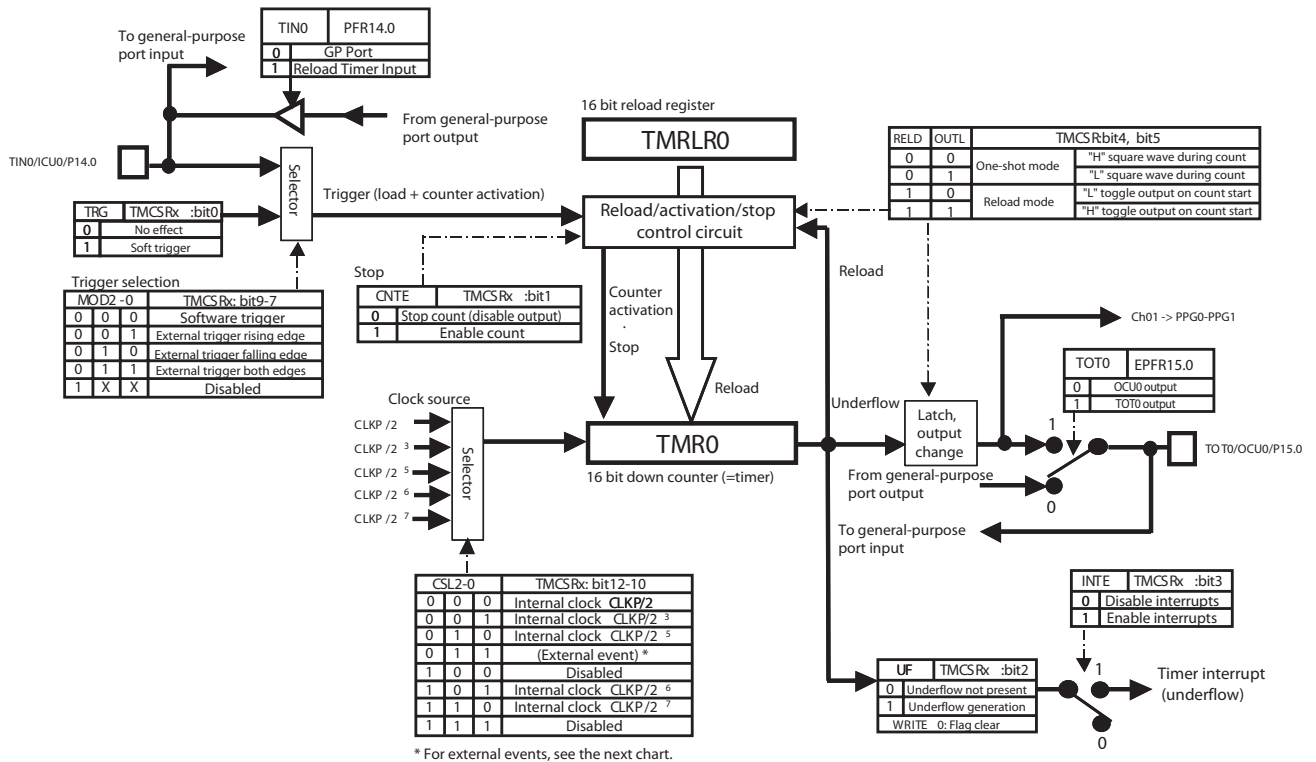
Count active edge: When in external event mode, choose from 3 types.

- External trigger (rising /falling/both edges)
Interrupt: Interrupt generated by underflow
- PPG activation trigger source:
 - Reload timer 0 : PPG0, PPG1
 - Reload timer 1 : PPG2, PPG3
 - Reload timer 2 : PPG4, PPG5
 - Reload timer 3 : PPG6, PPG7
 - Reload timer 4 : PPG8, PPG9
 - Reload timer 5 : PPG10, PPG11
 - Reload timer 6 : PPG12, PPG13
 - Reload timer 7 : PPG14, PPG15
 - Reload timer 8 : PPG16, PPG17
 - Reload timer 9 : PPG18, PPG19
 - Reload timer 10 : PPG20, PPG21
 - Reload timer 11 : PPG22, PPG23
 - Reload timer 12 : PPG24, PPG25
 - Reload timer 13 : PPG26, PPG27
 - Reload timer 14 : PPG28, PPG29
 - Reload timer 15 : PPG30, PPG31
- A/D converter activation trigger source (Reload timer 7 : A/D)
If 2 ADC macros are implemented, Reload timer 7 supplies both ADCs.

38.3. Configuration

Figure 38.3-1 Configuration Diagram of the Reload Timer (Internal Clock Count)

Reload Timer 0 (Internal clock count)



16 bit reload register

TMRLRO

Reload/activation/stop control circuit

TMRO
16 bit down counter

Event source

Selector

Active edge

MOD2-0 | **TMCSRbit9 -7**

0	0	0	-----
0	0	1	Rising edge
0	1	0	Falling edge
0	1	1	Both edges
1	X	X	Disabled

CSL2-0 | **TMCSRbit12-10**

0	1	1	External event *
---	---	---	------------------

* For internal clock, see the previous chart.

TRG | **TMCSRbit0**

0	No effect
1	Soft trigger

CNTL | **TMCSRbit1**

0	Stop count
1	Enable count

TINO | **PFR14.0**

0	GP Port
1	Reload Timer Input

RELD | **OUTL** | **TMCSR: bit4, bit5**

0	0	One-shot mode	"H" square wave during count
0	1		"L" square wave during count
1	0	Reload mode	"L" toggle output on count start
1	1		"H" toggle output on count start

TOT0 | **EPFR15.0**

0	OC0U output
1	TOT0 output

INTL | **TMCSRbit3**

0	Disable interrupts
1	Enable interrupts

UF | **TMCSRbit2**

0	Underflow not present
1	Underflow generation

WRITE | **0: Flag clear**

Ch01 -> PPG0-PPG1

TOT0/OC0U/P15.0

Timer interrupt (underflow)

Figure 38.3-3 Register List

Reload Timer 0

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0001B6H	---	---	---	CSL2	CSL1	CSL0	MOD2	MOD1	MOD0	---	OUTL	RELD	INTE	UF	CNTE	TRG	TMCSR0	(Reload timer control status 0)		
0001B2H																		TMR0	(Down counter 0)	
0001B0H																			TMRLR0	(Reload 0)

Bit	7	6	5	4	3	2	1	0		
000D8FH	OCU7	OCU6	OCU5	OCU4	OCU3	OCU2	OCU1	OCU0	PFR15	(Port function 15)
000DCFH	TOT7	TOT6	TOT5	TOT4	TOT3	TOT2	TOT1	TOT0	EPFR15	(Extra port function 15)

000448H	---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR08	(Reload timer 0 interrupt level)
---------	-----	-----	-----	------	------	------	------	------	-------	----------------------------------

0FFF7CH																		32Bits	(Interrupt vector #32)
---------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--------	------------------------

* For information about ICR register and interrupt vectors, see the section entitled "Interrupt Control".

Note: For information about ICR registers and interrupt vectors, see “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

MB91460 Series

38.4. Registers

38.4.1 TMCSR: Reload Timer Control Status Register

The TMCSR controls the module operation and the interrupts. It also indicates the status of the reload timer.

- **TMCSR0** (Reload timer 0): Address: 001B6H (Access: Byte, Half-word)
- **TMCSR1** (Reload timer 1): Address: 001BEH (Access: Byte, Half-word)
- **TMCSR2** (Reload timer 2): Address: 001C6H (Access: Byte, Half-word)
- **TMCSR3** (Reload timer 3): Address: 001CEH (Access: Byte, Half-word)
- **TMCSR4** (Reload timer 4): Address: 001D6H (Access: Byte, Half-word)
- **TMCSR5** (Reload timer 5): Address: 001DEH (Access: Byte, Half-word)
- **TMCSR6** (Reload timer 6): Address: 001E6H (Access: Byte, Half-word)
- **TMCSR7** (Reload timer 7): Address: 001EEH (Access: Byte, Half-word)
- **TMCSR8** (Reload timer 8): Address: 00596H (Access: Byte, Half-word)
- **TMCSR9** (Reload timer 9): Address: 0059EH (Access: Byte, Half-word)
- **TMCSR10** (Reload timer 10): Address: 005A6H (Access: Byte, Half-word)
- **TMCSR11** (Reload timer 11): Address: 005AEH (Access: Byte, Half-word)
- **TMCSR12** (Reload timer 12): Address: 005B6H (Access: Byte, Half-word)
- **TMCSR13** (Reload timer 13): Address: 005BEH (Access: Byte, Half-word)
- **TMCSR14** (Reload timer 14): Address: 005C6H (Access: Byte, Half-word)
- **TMCSR15** (Reload timer 15): Address: 005CEH (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	bit
–	–	–	CSL2	CSL1	CSL0	MOD2	MOD1	
–	–	–	0	0	0	0	0	Initial Value
RX/WX	RX/WX	RX/WX	R0/WX	R/W	R/W	R/W0	R/W	Attribute
×	×	×	×	×	×	×	×	Rewrite during operation

7	6	5	4	3	2	1	0	bit
MOD0	–	OUTL	RELD	INTE	UF	CNTE	TRG	
0	–	0	0	0	0	0	0	Initial Value
R/W	RX/WX	R/W	R/W	R/W	R(RM1), W	R/W	R0/W	Attribute
×	–	×	×	×	0	0	0	Rewrite during operation

(0: can be rewritten, x: cannot be rewritten)

(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit15-14: Undefined

Writing has no effect on the operation. The read value is “0”.

- bit13: Undefined (reload timer 0 - reload timer 2)

Always write “0”. The read value is “0”.

- bit12-10: Count clock selection

CLKP: peripheral clock

CSL2	CSL1	CSL0	Count clock	Remarks
0	0	0	Internal clock CLKP/2	
0	0	1	Internal clock CLKP/8	
0	1	0	Internal clock CLKP/32	
0	1	1	External event (external clock)	

1	0	1	Internal clock CLKP/64	
1	1	0	Internal clock CLKP/128	

Notes: Depending on whether an internal clock or an external event is selected, the meaning of the operation mode selection bit (MOD[2:0]) changes.

- bit9-7: Operation mode selection

Reload trigger when internal clock is selected

MOD2	MOD1	MOD0	Reload trigger
0	0	0	Software trigger
0	0	1	External trigger (rising edge)
0	1	0	External trigger (falling edge)
0	1	1	External trigger (both edges)

When the selected reload trigger is input, the value of reload register TMRLR is loaded to the down counter and the count operation is started.

Count trigger when external event is selected

MOD2	MOD1	MOD0	Count trigger
0	0	0	-----
0	0	1	External trigger (rising edge)
0	1	0	External trigger (falling edge)
0	1	1	External trigger (both edges)

Counts an external event using the selected count trigger.

Always set MOD2 to "0". The read value is the written value.

- bit6: Undefined

Writing has no effect on the operation. The read value is "0".

- bit5: Output level setting

OUTL	One-shot mode (RELD="0")	Reload mode (RELD="1")
0	During count "H" square wave	During count start "L" toggle output
1	During count "L" square wave	During count start "H" toggle output

- During one-shot mode, a pulse is output during the count, and during reload mode a toggle is output.

- For output level setting bit "0" and "1" the output level is reversed.

- bit4: Enable reload

RELD	Enable reload
0	One-shot mode (reload disabled)
1	Reload mode (reload enabled)

- In reload mode, down counter underflow (0000H -> FFFFH) causes the value set to reload register (TMRLR) to be loaded to the down counter, and the count operation continues.

- In one-shot mode, down counter underflow (0000H -> FFFFH) causes the count operation to stop.

- bit3: Enable Reload timer interrupt requests

INTE	Enable timer interrupt requests
0	Disable interrupt requests
1	Enable interrupt requests

When timer interrupt requests are enabled, and the timer interrupt flag (UF) becomes "1" an interrupt request is

generated.

- bit2: Timer interrupt flag

UF	Timer interrupt request flag	
	When read	When write
0	Underflow not present	Clear interrupt requests
1	Underflow present	No effect

Upon a down counter underflow (0000H -> FFFFH) generation, the timer interrupt flag is set to "1". If the interrupt request is enabled (INTE="1") an interrupt request is generated.

- bit1: Enable timer count

CNTE	Enable timer count
0	Stop count operation
1	Enable count operation (waiting for activation trigger)

If timer count is enabled and an activation trigger is generated, the count operation starts. The activation trigger can be a software trigger or an external trigger.

- bit0: Software trigger

TRG	Software trigger
0	No effect. (The read value is "0".)
1	Start count operation after data load.

If the count operation is enabled (CNTE="1") and the software trigger bit is set to "1", the value of the reload register (TMRLR) is loaded to the down counter and the count operation starts.

If the count operation is not enabled (CNTE="0"), the software trigger has no effect.

38.4.2 TMR: Timer Register

- **TMR0 (Reload timer 0): Address: 01B2H (Access: Half-word)**
- **TMR1 (Reload timer 1): Address: 01BAH (Access: Half-word)**
- **TMR2 (Reload timer 2): Address: 01C2H (Access: Half-word)**
- **TMR3 (Reload timer 3): Address: 01CAH (Access: Half-word)**
- **TMR4 (Reload timer 4): Address: 01D2H (Access: Half-word)**
- **TMR5 (Reload timer 5): Address: 01DAH (Access: Half-word)**
- **TMR6 (Reload timer 6): Address: 01E2H (Access: Half-word)**
- **TMR7 (Reload timer 7): Address: 01EAH (Access: Half-word)**
- **TMR8 (Reload timer 8): Address: 0592H (Access: Half-word)**
- **TMR9 (Reload timer 9): Address: 059AH (Access: Half-word)**
- **TMR10 (Reload timer 10): Address: 05A2H (Access: Half-word)**
- **TMR11 (Reload timer 11): Address: 05AAH (Access: Half-word)**
- **TMR12 (Reload timer 12): Address: 05B2H (Access: Half-word)**
- **TMR13 (Reload timer 13): Address: 05BAH (Access: Half-word)**
- **TMR14 (Reload timer 14): Address: 05C2H (Access: Half-word)**
- **TMR15 (Reload timer 15): Address: 05CAH (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial Value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial Value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

The reload timer count value can be read out through the timer register TMR.

Please perform the read out using half-word access.

38.4.3 TMRLR: Reload register

- **TMRLR0 (Reload timer 0): Address: 01B0H (Access: Half-word)**
- **TMRLR1 (Reload timer 1): Address: 01B8H (Access: Half-word)**
- **TMRLR2 (Reload timer 2): Address: 01C0H (Access: Half-word)**
- **TMRLR3 (Reload timer 3): Address: 01C8H (Access: Half-word)**
- **TMRLR4 (Reload timer 4): Address: 01D0H (Access: Half-word)**
- **TMRLR5 (Reload timer 5): Address: 01D8H (Access: Half-word)**
- **TMRLR6 (Reload timer 6): Address: 01E0H (Access: Half-word)**
- **TMRLR7 (Reload timer 7): Address: 01E8H (Access: Half-word)**
- **TMRLR8 (Reload timer 8): Address: 0590H (Access: Half-word)**
- **TMRLR9 (Reload timer 9): Address: 0598H (Access: Half-word)**
- **TMRLR10 (Reload timer 10): Address: 05A0H (Access: Half-word)**
- **TMRLR11 (Reload timer 11): Address: 05A8H (Access: Half-word)**
- **TMRLR12 (Reload timer 12): Address: 05B0H (Access: Half-word)**
- **TMRLR13 (Reload timer 13): Address: 05B8H (Access: Half-word)**

MB91460 Series

- **TMRLR14 (Reload timer 14): Address: 05C0H (Access: Half-word)**
- **TMRLR15 (Reload timer 15): Address: 05C8H (Access: Half-word)**

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial Value
RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial Value
RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	Attribute

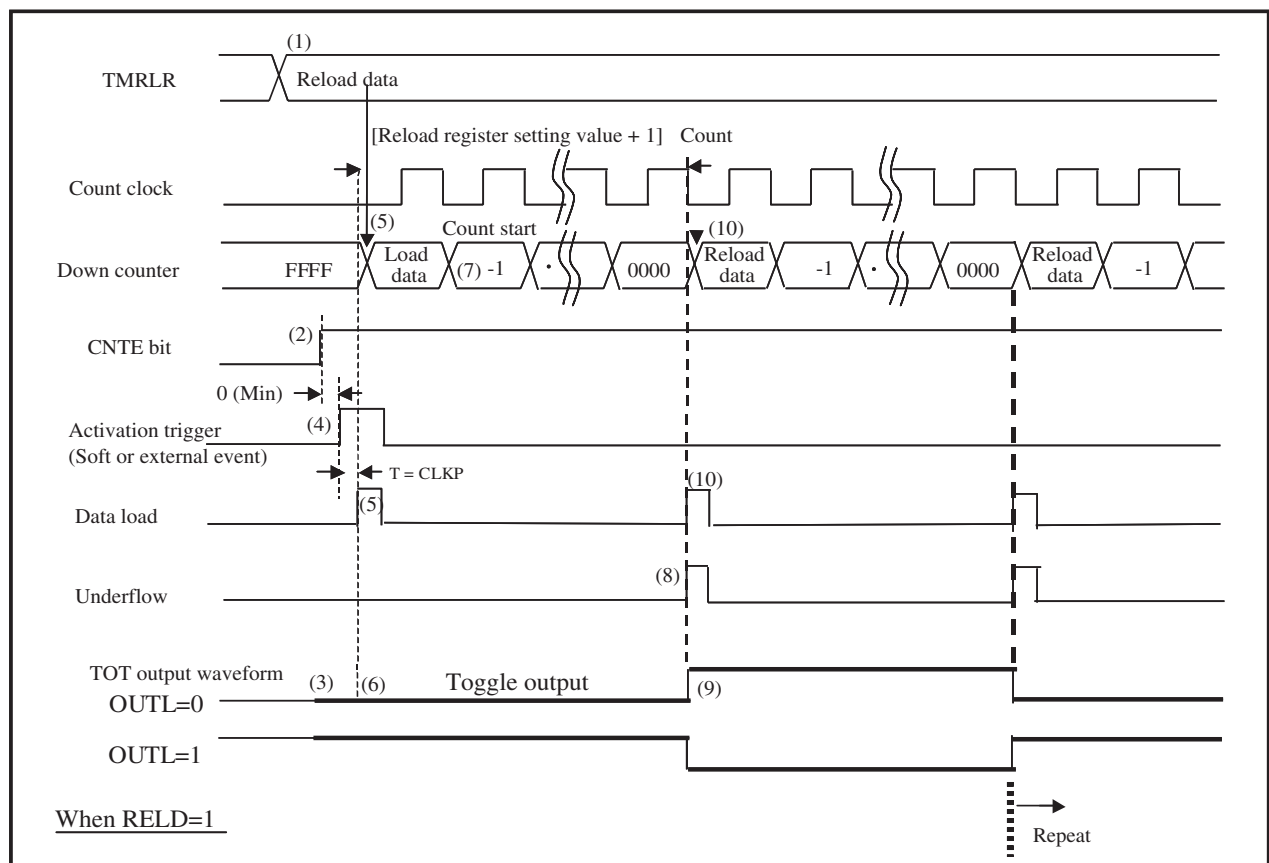
(For information on attributes, see “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

The reload value for the down counter is stored in reload register TMRLR.

Please write using half-word access.

38.5. Operation**38.5.1 Internal Clock/Reload Mode**

Figure 38.5-1 Operation of Reload Timer in reload mode (pulse with a 50% duty ratio is output)



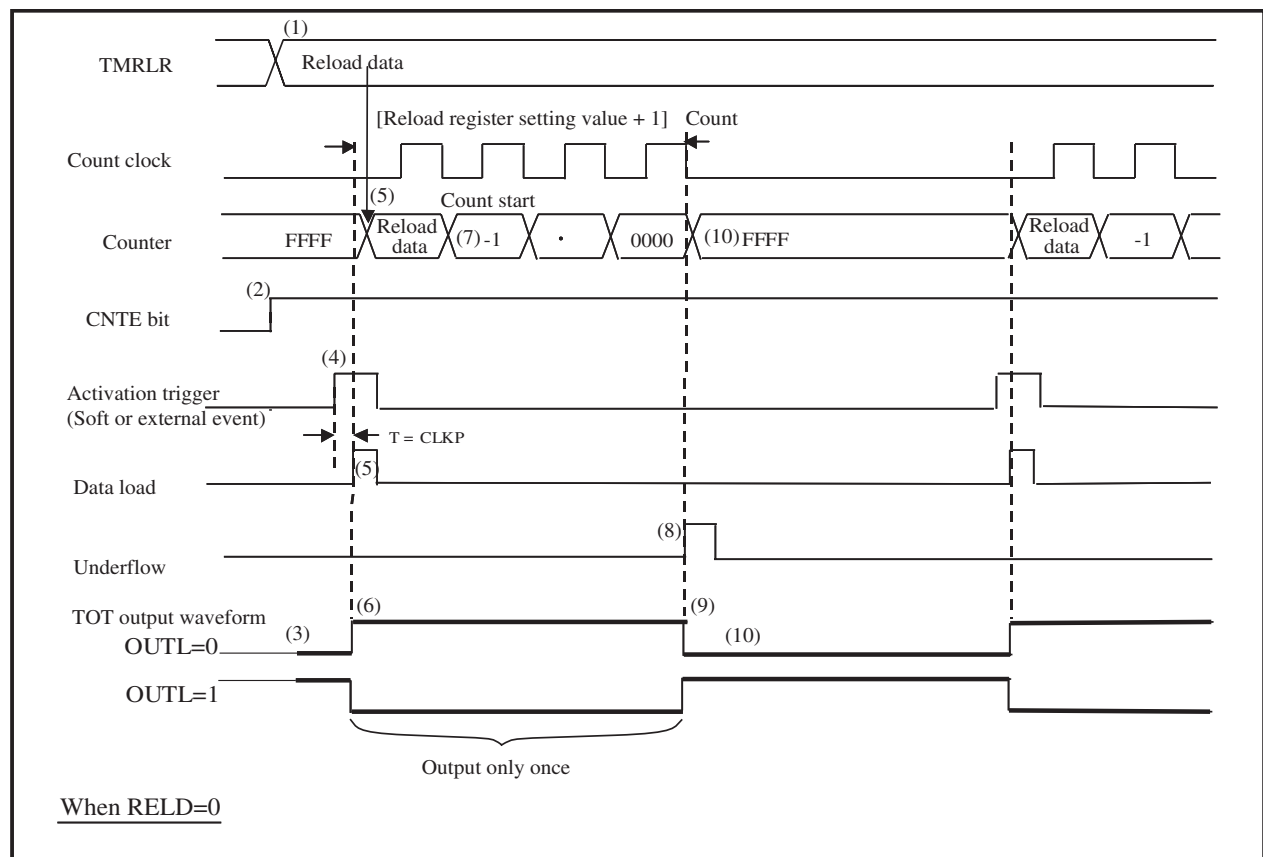
- (1) Set reload value to reload register
- (2) Enable reload timer count operation
- (3) TOT pin output

- (4) Generate reload trigger (activation): soft trigger or external event trigger
- (5) Load reload value
- (6) TOT toggle output start
- (7) Counter count down (internal clock synchronous)
- (8) Generate counter underflow
- (9) TOT pin output level reversal (toggle output)
- (10) Reload reload value
- (11) Repeat steps (7) to (10)

(See “38.8. Caution (Page No.940)”.)

38.5.2 Internal Clock/One-shot Mode

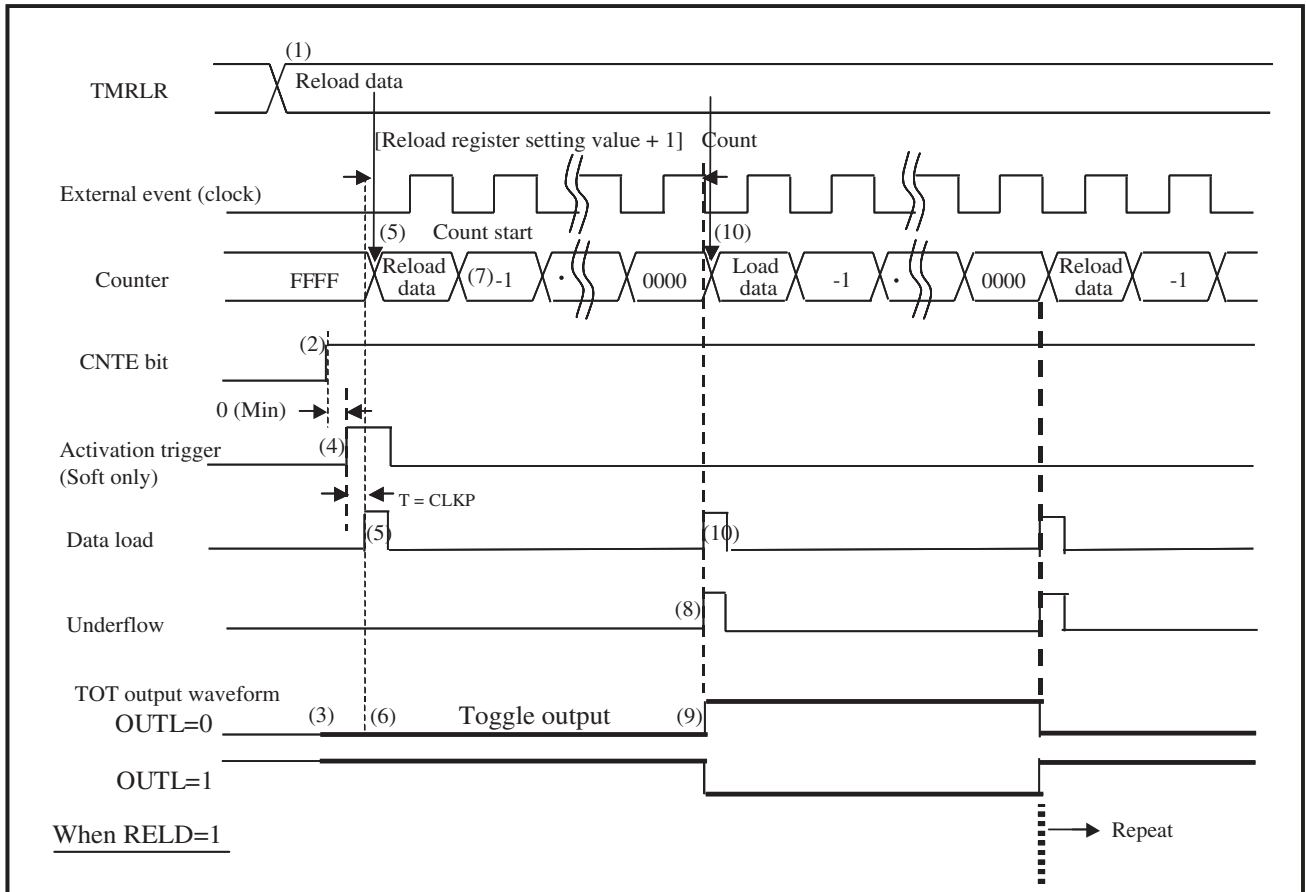
Figure 38.5-2 Operation of Reload Timer in one-shot mode (one-shot pulse is output)



- (1) Set reload value to reload register
- (2) Enable reload timer count operation
- (3) TOT pin output
- (4) Generate reload trigger (activation): soft trigger or external event trigger
- (5) Load reload value
- (6) Square wave output (during count, “H” output/OUTL=“0”)
- (7) Counter count down (internal clock synchronous)
- (8) Generate counter underflow
- (9) Return TOT pin output level
- (10) Count stop, wait for next activation trigger

MB91460 Series

(See “38.8. Caution (Page No.940)”.)

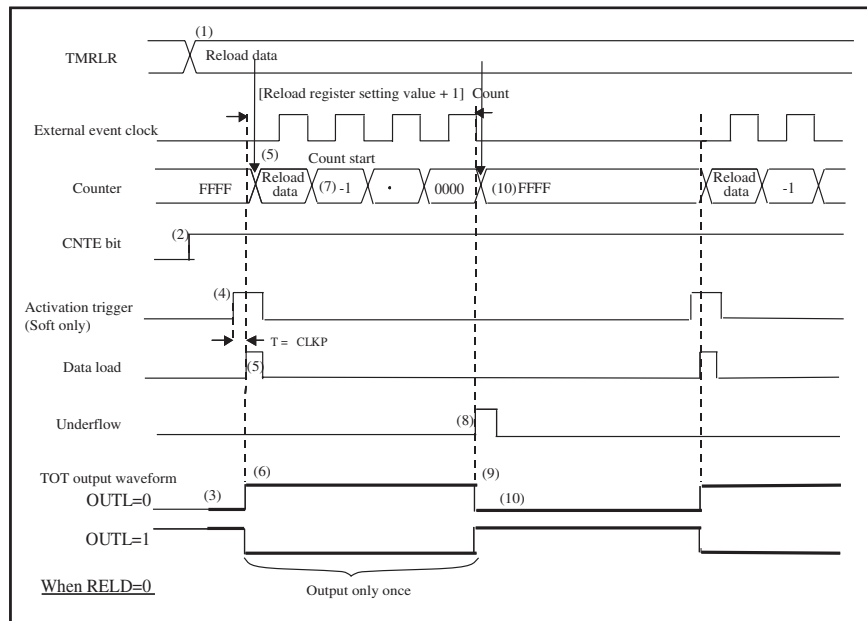
38.5.3 External Event Clock Reload Mode**Figure 38.5-3 Operation in external event clock reload mode (outputs a pulse with a 50% duty ratio)**

- (1) Set reload value to reload register
- (2) Enable reload timer count operation
- (3) TOT pin output
- (4) Generate reload trigger (activation): software trigger only
- (5) Load reload value
- (6) TOT pin output (initial value)
- (7) Counter count down (external event synchronous)
- (9) TOT pin output level reversal
- (10) Reload reload value
- (11) Repeat steps (6) to (9)

(See “38.8. Caution (Page No.940)”.)

38.5.4 External Event Clock/One-shot Mode

Figure 38.5-4 Operation in external event clock one-shot mode, (one-shot pulse is output)



- (1) Set reload value to reload register
- (2) Enable reload timer count operation
- (3) TOT pin output
- (4) Generate reload trigger (activation): soft trigger only
- (5) Load reload value
- (6) TOT pin output (during count, output "H"/OUTL="0")
- (7) Counter count down (via external events)
- (8) Generate counter underflow
- (9) TOT pin output reversal
- (10) Stop counter, wait for next activation trigger

Note: The first reload is delayed by a maximum of 1 T (T: count clock cycle).

38.5.5 Operation during Reset

A reset (reset on INITX signal, watchdog reset, software reset) will cause the registers in the reload timer to be initialized. The initial value of reload registers is indeterminate.

For detailed information on initial values, see the explanation of registers.

38.5.6 Operation during SLEEP state

Even after entering SLEEP state, the reload timer operation will continue.

38.5.7 Operation during STOP state

After entering STOP state, the operation of the reload timer stops.

MB91460 Series

38.5.8 Operation when Returning from STOP state

When returning from STOP state due to an external interrupt, the reload timer will continue with the state it had before the transition to STOP state.

When returning from STOP state due to a reset (INITX), the reload timer will return to the initial state (down counter stopped, no TOT pin output).

38.5.9 State Transition

The state of the counter is determined by the CNTE bit of the reload timer control register and the internal WAIT signal.

Valid states are:

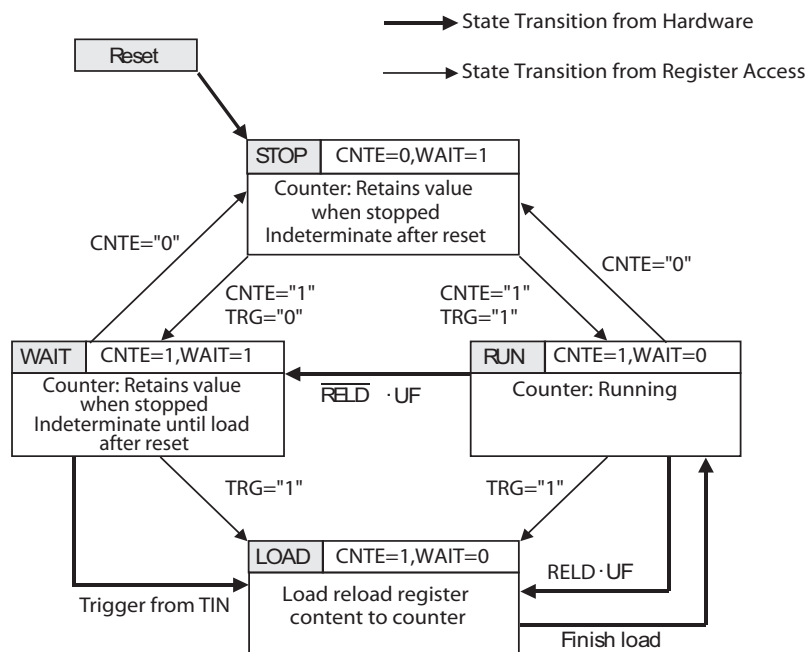
STOP state: Stopped (CNTE="0", WAIT="1")

WAIT state: Waiting for activation trigger (CNTE="1", WAIT="1")

RUN state : Count operation running (CNTE="1", WAIT="0")

LOAD state: Loading value to counter (CNTE="1", WAIT="0")

Figure 38.5-5 State Transition Diagram of the Reload Timer



38.6. Setting

Table 38.6-1 Settings necessary for internal clock operation

Setting	Setting Registers	Setting Procedure
Reload value settings	Reload (TMRLR0-TMRLR15)	See 38.7.1
Count clock selection (internal clock selection)	Reload timer control status (TMCSR0-TMCSR15)	See 38.7.2
Enable reload timer count operation		See 38.7.3
Mode selection (reload /one-shot)		See 38.7.4
Output reversal specification		See 38.7.5
Reload trigger selection (activation selection) Soft trigger External trigger (Rising edge/falling edge/both edges)		See 38.7.6 , see 38.7.7
TOT0 - TOT7 pin output *1	(Extra) port function register EPFR15 and PFR15	See 38.7.8
Generate activation trigger	—	See 38.7.10
Software trigger -> Software trigger bit setting	Reload timer control status (TMCSR0-TMCSR15)	
External trigger -> Input trigger to TIN pin	External input	

1. Reload timers 8 to 15 do not have a TOT output pin

Table 38.6-2 Settings necessary for external event operation

Setting	Setting Registers	Setting Procedure
Reload value setting	Reload (TMRLR0-TMRLR15)	See 38.7.1
Count clock selection (external event clock selection)	Reload timer control status (TMCSR0-TMCSR15)	See 38.7.2
Enable reload timer count operation		See 38.7.3
Mode selection (reload /one-shot)		See 38.7.4
Output reversal specification		See 38.7.5
External event clock active edge selection (Rising edge/falling edge/both edges)		See 38.7.7
TOT0 - TOT7 pin output *1	(Extra) port function register EPFR15 and PFR15	See 38.7.8
TIN0 - TIN15 pin external event input	Port function register (PFR14.0 - PFR14.7)	See 38.7.7
Generate activation trigger Software trigger -> Software trigger bit setting	Reload timer control status (TMCSR0-TMCSR15)	See 38.7.10

1. Reload timers 8 to 15 do not have a TOT output pin

MB91460 Series**Table 38.6-3 Items Necessary for Performing Reload Timer Interrupts**

Setting	Setting Registers	Setting Procedure
Reload timer interrupt vector Reload timer interrupt level setting	See “ Chapter 24 Interrupt Control (Page No.429) ”	See 38.7.11
Reload timer interrupt settings Interrupt clear Enable interrupt requests	Reload timer control status (TMCSR0-TMCSR15)	See 38.7.12

Table 38.6-4 Settings Necessary for Stopping the Reload Timer

Setting	Setting Registers	Setting Procedure
Reload timer stop bit setting	Reload timer control status (TMCSR0-TMCSR15)	See 38.7.13

38.7. Q & A

38.7.1 How to set the reload value

The reload value is set by the 16 bit reload registers TMRLR0-TMRLR15.

The equation for the values to be set is as follows.

- Formula

$$\text{TMRLR register value} = \{\text{reload interval/count clock}\} - 1$$
- Allowed Range

$$\text{TMRLR register value} = 0 \sim \text{FFFh} \text{ (65535)}$$

38.7.2 How to select the count clock

The count clock is chosen from the possibilities in the table below.

Selection is done via the count clock selection bit.

Table 38.7-1 TMCSR.CSL[2:0]

Count Clock	Counter clock selection bit			Count clock example		
	CSL2	CSL1	CSL0	When CLKP= 32MHz	When CLKP= 16MHz	When CLKP= 8MHz
CLKP/2	0	0	0	62.5ns	125ns	250ns
CLKP/8	0	0	1	250ns	500ns	1.0μs
CLKP/32	0	1	0	1.0μs	2.0μs	4.0μs
External event	0	1	1	Pulse width: 2/CLKP min		
CLKP/64	1	0	1	2.0μs	4.0μs	8.0μs
CLKP/128	1	1	0	4.0μs	8.0μs	16.0μs
Disabled *	1	0	0	-----		
	1	1	1	-----		

(*: See “38.8. Caution (Page No.940)”.)

38.7.3 How to enable/disable the reload timer count operation

Use the reload timer count enable bit (TMCSR.CNTE).

Control Details	RLT operation enable bit (CNTE)
To stop the reload timer	Set to “0”
To enable the reload timer's count operation	Set to “1”

If the timer should change from STOP state to RUN state, first set CNTE = 1 before setting TRG = 1. Setting TRG = 1 without setting CNTE first (or simultaneous) has no effect.

38.7.4 How to set the reload timer mode (reload/one-shot)

Use mode selection bit (TMCSR.RELD).




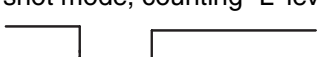
Operation Mode	Mode selection bit (RELD)
To set to one-shot mode	Set to “0”
To set to reload	Set to “1”

MB91460 Series

38.7.5 How to reverse the output level

The settings for the output level are detailed in the following table.

The setting is done via timer output level bit (TMCSR.OUTL).

Output level	Timer output level bit (OUTL)
Reload mode, Initial value "L" level output 	Set to "0"
Reload mode, initial value "H" level output (reversed) 	Set to "1"
One-shot mode, counting "H" level output 	Set to "0"
One-shot mode, counting "L" level output (reversed) 	Set to "1"

38.7.6 What are the types of triggers, and how to select them?

- Selection is done via the trigger selection bit (TMCSR.MOD[2:0]).

There are 4 types of reload triggers when an internal clock is selected.

Trigger	Trigger specification bit (MOD[2:0])
Software trigger (TRG bit set)	Set to "000"
External trigger from TINx pin (rising edge)	Set to "001"
External trigger from TINx pin (falling edge)	Set to "010"
External trigger from TINx pin (both edges)	Set to "011"
	"100", "101", "110", "111" are disabled *

Reload is repeated on down counter underflow.

(*: See "38.8. Caution (Page No.940)".)

- The reload trigger (activation) when no external event is selected is a software trigger.

38.7.7 What are the types of external event clock active edges and how to select them?

The setting is done via the trigger selection bit (TMCSR.MOD[1:0]).

There are three types of active edges.

Active edge	Trigger selection bit (MOD1-MOD0)
Rising edge	Set to "01"
Falling edge	Set to "10"
Both edges	Set to "11"

MOD2 settings have no meaning, no matter if they are set to "0" or "1".

38.7.8 How to make a pin a TOT output pin?

Write “1” to the TOT output selection bits (PFR15/EPFR15) to change the port to a TOT pin output.

Pin	Control bit	
TOT0 pin	PFR15.0 = ‘1’	EPFR15.0 = ‘1’
TOT1 pin	PFR15.1 = ‘1’	EPFR15.1 = ‘1’
TOT2 pin	PFR15.2 = ‘1’	EPFR15.2 = ‘1’
TOT3 pin	PFR15.3 = ‘1’	EPFR15.3 = ‘1’
TOT4 pin	PFR15.4 = ‘1’	EPFR15.4 = ‘1’
TOT5 pin	PFR15.5 = ‘1’	EPFR15.5 = ‘1’
TOT6 pin	PFR15.6 = ‘1’	EPFR15.6 = ‘1’
TOT7 pin	PFR15.7 = ‘1’	EPFR15.7 = ‘1’
TOT8 to TOT15 pin	Reload timers 8 to 15 do not have TOT pin	

38.7.9 How to make the TIN pin into an external event input pin, or an external trigger input pin?

Write “1” to the TIN input selection bits (PFR14) to change the port to a TIN pin input.

Pin	Control bit	
TIN0/8 pin	PFR14.0 = ‘1’	-
TIN1/9 pin	PFR14.1 = ‘1’	-
TIN2/10 pin	PFR14.2 = ‘1’	-
TIN3/11 pin	PFR14.3 = ‘1’	-
TIN4/12 pin	PFR14.4 = ‘1’	-
TIN5/13 pin	PFR14.5 = ‘1’	-
TIN6/14 pin	PFR14.6 = ‘1’	-
TIN7/15 pin	PFR14.7 = ‘1’	-

38.7.10 How to generate an activation trigger

- Generating a software trigger

The setting is done via the software trigger bit (TMCSR.TRG).

When the software trigger bit (TGR) is set to “1”, a trigger is generated.

To enable operation and activate at the same time, set the count permission bit (TMCSR.CNTE) and the soft trigger bit (TMCSR.TRG) simultaneously.

- Generating an external trigger

By inputting the edge specified by the trigger selection bit to the trigger pin corresponding to each reload timer, a trigger is generated.

Timer	Trigger pin
Reload timer 0 and Reload timer 8	TIN0/8
Reload timer 1 and Reload timer 9	TIN1/9
Reload timer 2 and Reload timer 10	TIN2/10
Reload timer 3 and Reload timer 11	TIN3/11
Reload timer 4 and Reload timer 12	TIN4/12
Reload timer 5 and Reload timer 13	TIN5/13
Reload timer 6 and Reload timer 14	TIN6/14
Reload timer 7 and Reload timer 15	TIN7/15

MB91460 Series

38.7.11 What are the interrupt-related registers?

The relationship between reload timer numbers, interrupt level, vector, control register, etc is outlined in the following table.

For details on interrupt level and interrupt vectors, see “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

	Interrupt vector (default)	Interrupt level setting bit (ICR[4:0])
Reload timer 0 + 8	#32 Address: 0FFF7Ch	Interrupt level register (ICR08) Address: 0448h
Reload timer 1 + 9	#33 Address: 0FFF78h	
Reload timer 2 + 10	#34 Address: 0FFF74h	Interrupt level register (ICR09) Address: 0449h
Reload timer 3 + 11	#35 Address: 0FFF70h	
Reload timer 4 + 12	#36 Address: 0FFF6Ch	Interrupt level register (ICR10) Address: 044Ah
Reload timer 5 + 13	#37 Address: 0FFF68h	
Reload timer 6 + 14	#38 Address: 0FFF64h	Interrupt level register (ICR11) Address: 044Bh
Reload timer 7 + 15	#39 Address: 0FFF60h	

Interrupt flag (TMCSR0.UF) ~ (TMCSR15.UF) is not automatically cleared, so before returning from interrupt processing, set the UF bit to “0” to reset it.

38.7.12 How to enable interrupts

Interrupt enable bit, interrupt flag

Enabling of interrupts is done via the interrupt request enable bit (TMCSR0.INTE) ~ (TMCSR15.INTE).

	Interrupt request enable bit (INTE)
To disable interrupt requests	Set to “0”
To enable interrupt requests	Set to “1”

Clearing of interrupts is done via the interrupt flag (TMCSR0.UF) ~ (TMCSR7.UF).

	Interrupt flag (UF)
To clear interrupts	Set to “0”

38.7.13 How to stop the reload timer?

This setting is done via the reload timer stop bit.

See “[38.7.3 How to enable/disable the reload timer count operation \(Page No.936\)](#)”.

38.8. Caution

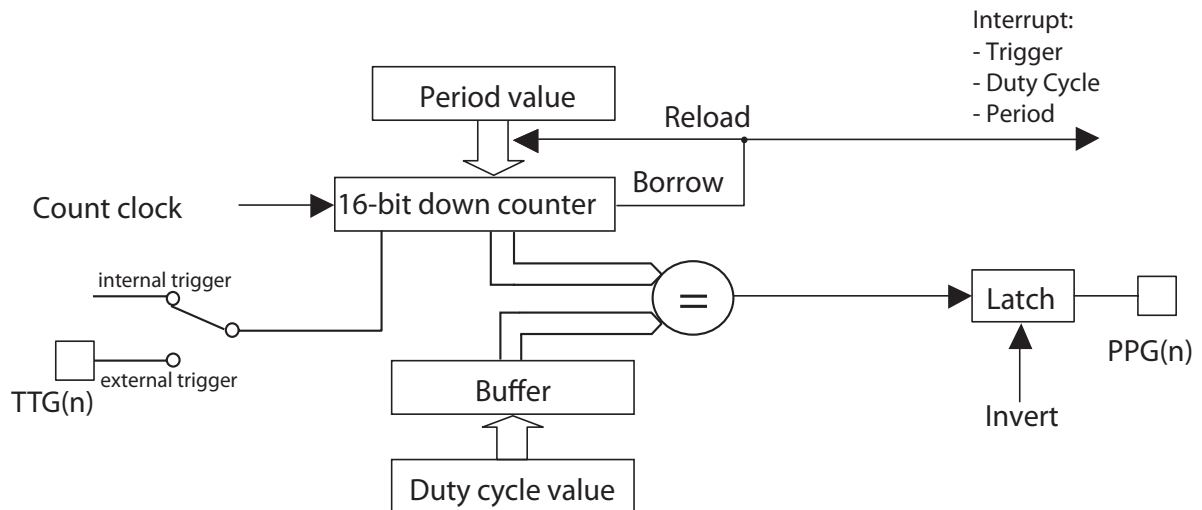
- Count source select bit (TMCSR.CSL[2:0]) settings not in the table: “100”, “111” are disabled. If they are set, disable the reload timer operation before resetting the count source select bit.
- Operation mode bit (TMCSR.MOD2) must be set to “0”. If it is set to “1”, disable the reload timer count operation before resetting it. Also the value written during read/modify/write access may be read.
- Control bits (Count source select, operation mode, reload permission) must not be rewritten during operation. If they are set during operation, disable the reload timer count operation before resetting them.
- From activation timing, it takes T cycle for the reload value to be loaded to the down counter. (Cycle = 1/CLKP, CLKP = peripheral clock)
- Reload timers 8 to 15 do not have a TOT output pin
- About output signal internal connections
 - Reload timer TOT0-TOT15 outputs are connected to the PPG0-PPG31 internal trigger inputs.
 - Reload timer TOT7 output is connected to the A/D converters 0+1 trigger input.
- Rewriting of the count clock selection bit (CSL[2:0]), operation mode selection bit (MOD[2:0]), output level setting bit (OUTL), reload permission bit (RELD), and timer interrupt request enable bit (INTE) should be done when the reload timer is stopped (TMCSR.CNTE=“0”).
- The internal prescaler should be already set when the timer count permission bit (TMCSR.CNTE) is set to “1”.
- If the setting of the interrupt flag and clearing the interrupt flag occur at the same time, the flag setting has higher priority and the clear operation will be made invalid.
- When writing to the reload register and a reload operation occur at the same time, the old data will be loaded to the counter. The new data will be loaded during the next reload operation.
- If the loading and counting of the timer register occur at the same time, the load (reload) operation has higher priority.
- If the counting should be enabled at the same time as the count operation should be started, the timer count enable bit (TMCSR.CNTE) and the software trigger bit (TMCSR.TRG) should both be set to “1”.

Chapter 39 Programmable Pulse Generator (PPG)

39.1. Overview

Programmable Pulse Generators (PPGs) are used to generate one-shot (rectangular wave) or pulse width modulation (PWM) output. The period and duty cycle of the PWM output and the duration of the one shot output are programmable by software.

Figure 39.1-1 Block diagram of PPG (Programmable Pulse Generator)



39.2. Features

Figure 39.2-1 Output waveforms: The PPGs can generate the following six types of output signals

- PWM waveform

Normal polarity:



Inverted polarity:



Figure 39.2-2 One-shot waveform (Rectangular wave)

Normal polarity:



Inverted polarity:



- Clamped output
Normal polarity: “L” Clamped output
Inverted polarity: “H” Clamped output
- Quantity: up to 8 groups (Output: 32 channels PPG0 - PPG31)

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Count clock: CLKP, CLKP/4, CLKP/16 or CLKP/64
- Period: Setting range = Duty cycle value ~ 65535 (specified with a 16-bit register)
Period = Count clock (PCSR register value + 1)
(Example) Count clock = 32MHz(31.25ns), PCSR value = 63999
Period = 31.25ns (63999+1) = 2ms
- Duty cycle: Setting range = 0 ~ Period value (specified with a 16-bit register)
Duty cycle = Count clock (PDUT register value + 1)
- Interrupt:
 - Software trigger
 - Counter borrow (period match)
 - Duty cycle match
 - Counter borrow (period match) or duty cycle match
- Activation trigger:
 - Software trigger
 - Internal triggers
 - Reload timer output 0 (TOT0) available as trigger for PPG0-PPG3
 - Reload timer output 1 (TOT1) available as trigger for PPG0-PPG3
 - Reload timer output 2 (TOT2) available as trigger for PPG4-PPG7
 - Reload timer output 3 (TOT3) available as trigger for PPG4-PPG7
 - Reload timer output 4 (TOT4) available as trigger for PPG8-PPG11
 - Reload timer output 5 (TOT5) available as trigger for PPG8-PPG11
 - Reload timer output 6 (TOT6) available as trigger for PPG12-PPG15
 - Reload timer output 7 (TOT7) available as trigger for PPG12-PPG15
 - Reload timer output 8 available as trigger for PPG16,PPG17
 - Reload timer output 9 available as trigger for PPG18,PPG19
 - Reload timer output 10 available as trigger for PPG20,PPG21
 - Reload timer output 11 available as trigger for PPG22,PPG23
 - Reload timer output 12 available as trigger for PPG24,PPG25
 - Reload timer output 13 available as trigger for PPG26,PPG27
 - Reload timer output 14 available as trigger for PPG28,PPG29
 - Reload timer output 15 available as trigger for PPG30,PPG31
 - External triggers
 - Port GP14_0 (ICU0, RLT0 ext-trig, TTG0/8/16/24) available as trigger for PPG0-PPG3/PPG8-PPG11/PPG16-PPG19/PPG24-PPG27
 - Port GP14_1 (ICU1, RLT1 ext-trig, TTG1/9/17/25) available as trigger for PPG0-PPG3/PPG8-PPG11/PPG16-PPG19/PPG24-PPG27
 - Port GP14_2 (ICU2, RLT2 ext-trig, TTG2/10/18/26) available as trigger for PPG0-PPG3/PPG8-PPG11/PPG16-PPG19/PPG24-PPG27
 - Port GP14_3 (ICU3, RLT3 ext-trig, TTG3/11/19/27) available as trigger for PPG0-PPG3/PPG8-PPG11/PPG16-PPG19/PPG24-PPG27
 - Port GP14_4 (ICU4, RLT4 ext-trig, TTG4/12/20/28) available as trigger for PPG4-PPG7/PPG12-PPG15/PPG20-PPG23/PPG28-PPG31
 - Port GP14_5 (ICU5, RLT5 ext-trig, TTG5/13/21/29) available as trigger for

PPG4-PPG7/PPG12-PPG15/PPG20-PPG23/PPG28-PPG31
Port GP14_6 (ICU6, RLT6 ext-trig, TTG6/14/22/30) available as trigger for
PPG4-PPG7/PPG12-PPG15/PPG20-PPG23/PPG28-PPG31
Port GP14_7 (ICU7, RLT7 ext-trig, TTG7/15/23/31) available as trigger for
PPG4-PPG7/PPG12-PPG15/PPG20-PPG23/PPG28-PPG31

PPG (0-3)

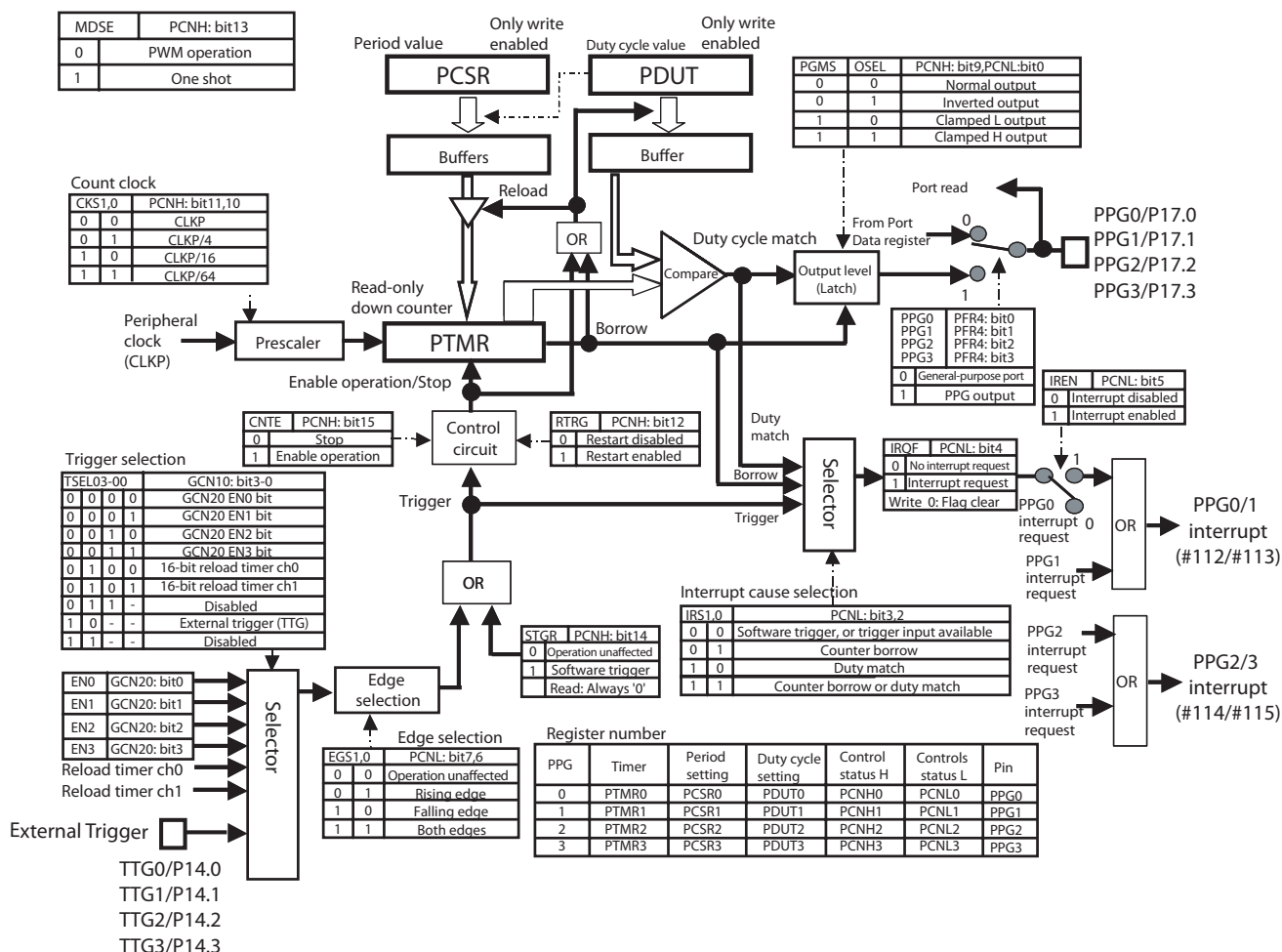


Figure 39.3-2 Register List

PPG0

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000110H	16 bits (Read Only)																	PTMR00	(PPG timer 0)
000112H	16 bits (Write Only)																	PCSR00	(PPG period setting 0)
000114H	16 bits (Write Only)																	PDUT00	(PPG duty setting 0)
000116H	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	---	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	---	OSEL	PCNH0,PCNL00		
000100H	TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00	GCN10 (General control 10)		
000103H	Bit	7	6	5	4	3	2	1	0										
		---	---	---	---	EN3	EN2	EN1	EN0	GCN20		(General control 20)							
000D91H	PPG7																		
000470H	---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR48		(Interrupt level PPG 0/1)								
0FFE3CH	32 bits																	PPG0	Interrupt vector (#112)

※For more information about the ICR register and interrupt vector, see the chapter entitled "Interrupt Control."

MB91460 Series

39.3.Configuration

PPG1

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000118H	16 bits (Read Only)																	PTMR01	(PPG timer 1)
00011AH	16 bits (Write Only)																	PCSR01	(PPG period setting 1)
00011CH	16 bits (Write Only)																	PDUT01	(PPG duty setting 1)
00011EH	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	---	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	---	OSEL	PCNH01,PCNL01	(PPG control status 1)	
000100H	TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00	GCN10	(General control 10)	
000103H	Bit	7	6	5	4	3	2	1	0										
		---	---	---	---	EN3	EN2	EN1	EN0	GCN20									(General control 20)
000D91H	PPG7	PPG6	PPG5	PPG4	PPG3	PPG2	PPG1	PPG0	PFR17										(Port function 17)
000470H	---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR48										(Interrupt level PPG 0/1)
0FFE38H	32 bits																	PPG1	Interrupt vector (#113)

※For more information about the ICR register and interrupt vector, see the chapter entitled "Interrupt Control."

PPG2

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000120H	16 bits (Read Only)																	PTMR02	(PPG timer 2)
000122H	16 bits (Write Only)																	PCSR02	(PPG period setting 2)
000124H	16 bits (Write Only)																	PDUT02	(PPG duty setting 2)
000126H	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	---	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	---	OSEL	PCNH02,PCNL02	(PPG control status 2)	
000100H	TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00	GCN10	(General control 10)	
000103H	---	---	---	---	EN3	EN2	EN1	EN0	GCN20										(General control 20)
000D91H	PPG7	PPG6	PPG5	PPG4	PPG3	PPG2	PPG1	PPG0	PFR17										(Port function 17)
000471H	---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR49										(Interrupt level PPG 2/3)
0FFE34H	32 bits																	PPG2	Interrupt vector (#114)

※For more information about the ICR register and interrupt vector, see the chapter entitled "Interrupt Control."

PPG3

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000128H	16 bits (Read Only)																	PTMR03	(PPG timer 3)
00012AH	16 bits (Write Only)																	PCSR03	(PPG period setting 3)
00012CH	16 bits (Write Only)																	PDUT03	(PPG duty setting 3)
00012EH	CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	---	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	---	OSEL	PCNH03,PCNL03	(PPG control status 3)	
000100H	TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00	GCN10	(General control 10)	
000103H	Bit	7	6	5	4	3	2	1	0										
	---	---	---	---	EN3	EN2	EN1	EN0	GCN20	(General control 20)									
000D91H	PPG7	PPG6	PPG5	PPG4	PPG3	PPG2	PPG1	PPG0	PFR17	(Port function 17)									
000471H	---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR49	(Interrupt level PPG 2/3)									
0FFE30H	32 bits																	PPG3	Interrupt vector (#115)

※For more information about the ICR register and interrupt vector, see the chapter entitled "Interrupt Control."

Note: For more information about the ICR register and interrupt vector, see “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

39.4. Registers

39.4.1 PCSR: PPG Period Setting Register

Controls the period of the PPG.

- **PCSR00 (PPG0): Address 0112h (Access: Half-word)**
- **PCSR01 (PPG1): Address 011Ah (Access: Half-word)**
- **PCSR02 (PPG2): Address 0122h (Access: Half-word)**
- **PCSR03 (PPG3): Address 012Ah (Access: Half-word)**
- **PCSR04 (PPG4): Address 0132h (Access: Half-word)**
- **PCSR05 (PPG5): Address 013Ah (Access: Half-word)**
- **PCSR06 (PPG6): Address 0142h (Access: Half-word)**
- **PCSR07 (PPG7): Address 014Ah (Access: Half-word)**
- **PCSR08 (PPG8): Address 0152h (Access: Half-word)**
- **PCSR09 (PPG9): Address 015Ah (Access: Half-word)**
- **PCSR10 (PPG10): Address 0162h (Access: Half-word)**
- **PCSR11 (PPG11): Address 016Ah (Access: Half-word)**
- **PCSR12 (PPG12): Address 0332h (Access: Half-word)**
- **PCSR13 (PPG13): Address 033Ah (Access: Half-word)**
- **PCSR14 (PPG14): Address 0342h (Access: Half-word)**
- **PCSR15 (PPG15): Address 034Ah (Access: Half-word)**
- **PCSR16 (PPG16): Address 0512h (Access: Half-word)**
- **PCSR17 (PPG17): Address 0518h (Access: Half-word)**
- **PCSR18 (PPG18): Address 0522h (Access: Half-word)**
- **PCSR19 (PPG19): Address 0528h (Access: Half-word)**
- **PCSR20 (PPG20): Address 0532h (Access: Half-word)**
- **PCSR21 (PPG21): Address 0538h (Access: Half-word)**
- **PCSR22 (PPG22): Address 0542h (Access: Half-word)**
- **PCSR23 (PPG23): Address 0548h (Access: Half-word)**
- **PCSR24 (PPG24): Address 0552h (Access: Half-word)**
- **PCSR25 (PPG25): Address 0558h (Access: Half-word)**
- **PCSR26 (PPG26): Address 0562h (Access: Half-word)**
- **PCSR27 (PPG27): Address 0568h (Access: Half-word)**
- **PCSR28 (PPG28): Address 0572h (Access: Half-word)**
- **PCSR29 (PPG29): Address 0578h (Access: Half-word)**
- **PCSR30 (PPG30): Address 0582h (Access: Half-word)**
- **PCSR31 (PPG31): Address 0588h (Access: Half-word)**

15	14	13	12	11	10	9	8	Bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute ¹

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute ¹

1. Read access is available on MB91FV460B, MB91F467P and MB91F467E only.

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- The PPG Period Setting registers are buffered. Transfers from the corresponding buffers to the 16-bit down counter take place automatically at counter underflow.
- After the PPG Period Setting registers have been written, be sure to set PPG Duty cycle Setting registers PDUT.
- Always access the PPG Period Setting registers in a half-word (16-bit) format.
(See “[39.8. Caution \(Page No.974\)](#)”.)
- Read access returns the original PCSR register (not the buffered value).

39.4.2 PDUT: PPG Duty cycle Setting Register

Sets the duty cycle of the PPG output waveform.

- PDUT00 (PPG0): Address 0114h (Access: Half-word)
- PDUT01 (PPG1): Address 011Ch (Access: Half-word)
- PDUT02 (PPG2): Address 0124h (Access: Half-word)
- PDUT03 (PPG3): Address 012Ch (Access: Half-word)
- PDUT04 (PPG4): Address 0134h (Access: Half-word)
- PDUT05 (PPG5): Address 013Ch (Access: Half-word)
- PDUT06 (PPG6): Address 0144h (Access: Half-word)
- PDUT07 (PPG7): Address 014Ch (Access: Half-word)
- PDUT08 (PPG8): Address 0154h (Access: Half-word)
- PDUT09 (PPG9): Address 015Ch (Access: Half-word)
- PDUT10 (PPG10): Address 0164h (Access: Half-word)
- PDUT11 (PPG11): Address 016Ch (Access: Half-word)
- PDUT12 (PPG12): Address 0334h (Access: Half-word)
- PDUT13 (PPG13): Address 033Ch (Access: Half-word)
- PDUT14 (PPG14): Address 0344h (Access: Half-word)
- PDUT15 (PPG15): Address 034Ch (Access: Half-word)
- PDUT16 (PPG16): Address 0514h (Access: Half-word)
- PDUT17 (PPG17): Address 051Ch (Access: Half-word)
- PDUT18 (PPG18): Address 0524h (Access: Half-word)
- PDUT19 (PPG19): Address 052Ch (Access: Half-word)
- PDUT20 (PPG20): Address 0534h (Access: Half-word)
- PDUT21 (PPG21): Address 053Ch (Access: Half-word)
- PDUT22 (PPG22): Address 0544h (Access: Half-word)
- PDUT23 (PPG23): Address 054Ch (Access: Half-word)
- PDUT24 (PPG24): Address 0554h (Access: Half-word)
- PDUT25 (PPG25): Address 055Ch (Access: Half-word)
- PDUT26 (PPG26): Address 0564h (Access: Half-word)
- PDUT27 (PPG27): Address 056Ch (Access: Half-word)
- PDUT28 (PPG28): Address 0574h (Access: Half-word)
- PDUT29 (PPG29): Address 057Ch (Access: Half-word)
- PDUT30 (PPG30): Address 0584h (Access: Half-word)
- PDUT31 (PPG31): Address 058Ch (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute ¹

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R, W	R, W	R, W	R, W	R, W	R, W	R, W	R, W	Attribute ¹

1. Read access is available on MB91FV460B, MB91F467P and MB91F467E only.

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- The PPG Duty cycle Setting registers are buffered.
- Set a value smaller than the setting of PPG Period Setting register PCSR in a PPG Duty cycle Setting register. (See “[39.8. Caution \(Page No.974\)](#)”.)
- If the same value as set in PPG Period Setting register PCSR is set in a PPG Duty cycle Setting register,
 - The output pin level is always “H” at normal polarity time. (OSEL=”0”)
 - The output pin level is always “L” at inverted polarity time. (OSEL=”1”)

(The OSEL bit is an output polarity specification bit of the PPG control register PCN.)
- Always access the PPG Duty cycle Setting registers in a half-word (16-bit) format.
(See “[39.8. Caution \(Page No.974\)](#)”.)
- Read access returns the original PDUT register (not the buffered value).

39.4.3 PCN: PPG Control Status register

Controls the operations and status of PPGs.

- **PCN00 (PPG0): Address 0116h (Access: Byte, Half-word)**
- **PCN01 (PPG1): Address 011Eh (Access: Byte, Half-word)**
- **PCN02 (PPG2): Address 0126h (Access: Byte, Half-word)**
- **PCN03 (PPG3): Address 012Eh (Access: Byte, Half-word)**
- **PCN04 (PPG4): Address 0136h (Access: Byte, Half-word)**
- **PCN05 (PPG5): Address 013Eh (Access: Byte, Half-word)**
- **PCN06 (PPG6): Address 0146h (Access: Byte, Half-word)**
- **PCN07 (PPG7): Address 014Eh (Access: Byte, Half-word)**
- **PCN08 (PPG8): Address 0156h (Access: Byte, Half-word)**
- **PCN09 (PPG9): Address 015Eh (Access: Byte, Half-word)**
- **PCN10 (PPG10): Address 0166h (Access: Byte, Half-word)**
- **PCN11 (PPG11): Address 016Eh (Access: Byte, Half-word)**
- **PCN12 (PPG12): Address 0336h (Access: Byte, Half-word)**
- **PCN13 (PPG13): Address 033Eh (Access: Byte, Half-word)**
- **PCN14 (PPG14): Address 0346h (Access: Byte, Half-word)**
- **PCN15 (PPG15): Address 034Eh (Access: Byte, Half-word)**
- **PCN16 (PPG16): Address 0516h (Access: Byte, Half-word)**
- **PCN17 (PPG17): Address 051Eh (Access: Byte, Half-word)**
- **PCN18 (PPG18): Address 0526h (Access: Byte, Half-word)**

MB91460 Series

- PCN19 (PPG19): Address 052Eh (Access: Byte, Half-word)
- PCN20 (PPG20): Address 0536h (Access: Byte, Half-word)
- PCN21 (PPG21): Address 053Eh (Access: Byte, Half-word)
- PCN22 (PPG22): Address 0546h (Access: Byte, Half-word)
- PCN23 (PPG23): Address 054Eh (Access: Byte, Half-word)
- PCN24 (PPG24): Address 0556h (Access: Byte, Half-word)
- PCN25 (PPG25): Address 055Eh (Access: Byte, Half-word)
- PCN26 (PPG26): Address 0566h (Access: Byte, Half-word)
- PCN27 (PPG27): Address 056Eh (Access: Byte, Half-word)
- PCN28 (PPG28): Address 0576h (Access: Byte, Half-word)
- PCN29 (PPG29): Address 057Eh (Access: Byte, Half-word)
- PCN30 (PPG30): Address 0586h (Access: Byte, Half-word)
- PCN31 (PPG31): Address 058Eh (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	–	
0	0	0	0	0	0	0	X	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O	x	x	x	x	x	–	Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS1	EGS0	IREN	IRQF	IRS1	IRS0	–	OSEL	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R(RM1), W	R/W	R/W	RX/WX	R/W	Attribute
x	x	O	O	x	x	–	x	Rewrite during operation

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

O: Rewritable, x: Not writable (See “[39.8. Caution \(Page No.974\)](#)”).

- Bit 15: Timer enable operation

CNTE	Operation
0	Operation disabled
1	Operation enabled

This bit enables the operation of the PPG.

- Bit 14: Software trigger

STGR	Operation
0	The operation is unaffected by writing (The read value always equals “0”).
1	Software trigger activation

When the Software Trigger bit is set to “1”, a software trigger is generated to activate the PPG, separately from the generation of an internal trigger (EN bit, reload timer output).

- Bit 13: Mode selection

MDSE	Mode
0	PWM operation
1	One-shot operation

- When the Mode Selection bit is set to “0”, a PWM operation is enabled to generate pulses in sequence.
- When the Mode Selection bit is set to “1”, pulse output takes place only once.
- Bit 12: Restart enable

RTRG	Operation
0	Disable restart.
1	Enable restart.

When the Enable Restart bit is set to “1”, a trigger (software/internal) leads to a restart of the PPG, (even if the PPG is enabled (CNTE = “1”). If the Enable Restart bit is set to “0” the trigger has no effect on the function of the PPG.

- Bits 11-10: Counter clock selection

CKS1	CKS0	Down Counter Count Clock Selection
0	0	Peripheral clock (CLKP)
0	1	Peripheral clock divided by 4
1	0	Peripheral clock divided by 16
1	1	Peripheral clock divided by 64

- Bit 9: PPG output mask selection

PGMS	Operation
0	No output mask
1	Output mask (Output “L” level latched: OSEL=“0”)

- When the PPG Output Mask Selection bit is set to “1”, the PPG output can be clamped at “L” or “H” regardless of the mode, period, and duty cycle settings.
- The output level can be specified using the Output Polarity Specification bit (PCN.OSEL).
- Bit 8: Undefined. The operation is unaffected by writing. The read value is indeterminate.
- Bits 7-6: Trigger input edge selection

EGS1	EGS0	Selected Edge
0	0	The operation is unaffected by writing.
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising edge, or, falling edge)

Select an edge to trigger the activation of the trigger input selected with the Trigger Specification bits using the Trigger Input Edge Selection bit (EGS[1:0]).

(GCN10[15:12]), (GCN10[11:8]), (GCN10[7:4]), and (GCN10[3:0]) of PPG3 to PPG0,
 (GCN11[15:12]), (GCN11[11:8]), (GCN11[7:4]), and (GCN11[3:0]) of PPG7 to PPG4,
 (GCN12[15:12]), (GCN12[11:8]), (GCN12[7:4]), and (GCN12[3:0]) of PPG11 to PPG8,
 (GCN13[15:12]), (GCN13[11:8]), (GCN13[7:4]), and (GCN13[3:0]) of PPG15 to PPG12,
 (GCN14[15:12]), (GCN14[11:8]), (GCN14[7:4]), and (GCN14[3:0]) of PPG16 to PPG19,
 (GCN15[15:12]), (GCN15[11:8]), (GCN15[7:4]), and (GCN15[3:0]) of PPG20 to PPG23,
 (GCN16[15:12]), (GCN16[11:8]), (GCN16[7:4]), and (GCN16[3:0]) of PPG24 to PPG27,
 (GCN17[15:12]), (GCN17[11:8]), (GCN17[7:4]), and (GCN17[3:0]) of PPG28 to PPG31,

- Bit 5: Interrupt request enable

IREN	Operation
0	Interrupt request disabled
1	Interrupt request enabled

MB91460 Series

- Bit 4: interrupt flag

IRQF	Read Operation	Write Operation
0	No interrupt present	Clear the Interrupt flag.
1	Interrupt present	No effect.

If the Interrupt flag (IRQF) is set to “1” and writing “0” to the flag take place at the same time, the setting of the Interrupt flag (IRQF=“1”) has higher priority.

- Bit 3-2: Interrupt cause selection

IRS1	IRS0	Selection
0	0	Software trigger, or, trigger input
0	1	Counter borrow
1	0	The counter matches the duty cycle value.
1	1	Counter borrow, or the counter equals the duty cycle value.

- Select the operation in which to generate an interrupt request.
- Bit 1: Undefined.The operation is unaffected by writing. The read value is indeterminate.
- Bit 0: PPG output polarity specification

OSEL	Operation
0	Normal polarity
1	Inverted polarity

When the PPG Output Mask Selection bit (PCN.PGMS) has been set to “1”, if the Output Polarity Specification bit (OSEL) is set to “0”, the output is clamped at “L”; if the Output Polarity Specification bit is set to “1”, the output is clamped at “H”.

39.4.4 GCN1: General Control register 1

Selects a trigger input to PPG0-PPG3, PPG4-PPG7, PPG8-PPG11 and PPG12-PPG15.

- GCN10 (PPG0-PPG3): Address 0100h (Access: Half-word)
- GCN11 (PPG4-PPG7): Address 0104h (Access: Half-word)
- GCN12 (PPG8-PPG11): Address 0108h (Access: Half-word)
- GCN13 (PPG12-PPG15): Address 0320h (Access: Half-word)
- GCN14 (PPG16-PPG19): Address 0500h (Access: Half-word)
- GCN15 (PPG20-PPG23): Address 0504h (Access: Half-word)
- GCN16 (PPG24-PPG27): Address 0505h (Access: Half-word)
- GCN17 (PPG28-PPG31): Address 050Ch (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
TSEL33	TSEL32	TSEL31	TSEL30	TSEL23	TSEL22	TSEL21	TSEL20	
0	0	1	1	0	0	1	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	Bit
TSEL13	TSEL12	TSEL11	TSEL10	TSEL03	TSEL02	TSEL01	TSEL00	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

PPG28-PPG31:

- GCN17: Bits 15-12 (TSEL3[3:0]) PPG31 trigger specification
- GCN17: Bits 11-8 (TSEL2[3:0]) PPG30 trigger specification
- GCN17: Bits 7-4 (TSEL1[3:0]) PPG29 trigger specification
- GCN17: Bits 3-0 (TSEL0[3:0]) PPG28 trigger specification

PPG24-PPG27:

- GCN16: Bits 15-12 (TSEL3[3:0]) PPG27 trigger specification
- GCN16: Bits 11-8 (TSEL2[3:0]) PPG26 trigger specification
- GCN16: Bits 7-4 (TSEL1[3:0]) PPG25 trigger specification
- GCN16: Bits 3-0 (TSEL0[3:0]) PPG24 trigger specification

PPG20-PPG23:

- GCN15: Bits 15-12 (TSEL3[3:0]) PPG23 trigger specification
- GCN15: Bits 11-8 (TSEL2[3:0]) PPG22 trigger specification
- GCN15: Bits 7-4 (TSEL1[3:0]) PPG21 trigger specification
- GCN15: Bits 3-0 (TSEL0[3:0]) PPG20 trigger specification

PPG16-PPG19:

- GCN14: Bits 15-12 (TSEL3[3:0]) PPG19 trigger specification
- GCN14: Bits 11-8 (TSEL2[3:0]) PPG18 trigger specification
- GCN14: Bits 7-4 (TSEL1[3:0]) PPG17 trigger specification
- GCN14: Bits 3-0 (TSEL0[3:0]) PPG16 trigger specification

PPG12-PPG15:

- GCN13: Bits 15-12 (TSEL3[3:0]) PPG15 trigger specification
- GCN13: Bits 11-8 (TSEL2[3:0]) PPG14 trigger specification
- GCN13: Bits 7-4 (TSEL1[3:0]) PPG13 trigger specification
- GCN13: Bits 3-0 (TSEL0[3:0]) PPG12 trigger specification

PPG8-PPG11:

- GCN12: Bits 15-12 (TSEL3[3:0]) PPG11 trigger specification
- GCN12: Bits 11-8 (TSEL2[3:0]) PPG10 trigger specification
- GCN12: Bits 7-4 (TSEL1[3:0]) PPG9 trigger specification
- GCN12: Bits 3-0 (TSEL0[3:0]) PPG8 trigger specification

PPG4-PPG7:

- GCN11: Bits 15-12 (TSEL3[3:0]) PPG7 trigger specification
- GCN11: Bits 11-8 (TSEL2[3:0]) PPG6 trigger specification
- GCN11: Bits 7-4 (TSEL1[3:0]) PPG5 trigger specification
- GCN11: Bits 3-0 (TSEL0[3:0]) PPG4 trigger specification

PPG0-PPG3:

- GCN10: Bits 15-12 (TSEL3[3:0]) PPG3 trigger specification
- GCN10: Bits 11-8 (TSEL2[3:0]) PPG2 trigger specification
- GCN10: Bits 7-4 (TSEL1[3:0]) PPG1 trigger specification
- GCN10: Bits 3-0 (TSEL0[3:0]) PPG0 trigger specification

TSEL				Activation trigger specification
0	0	0	0	EN0 bit (GCN2 register)
0	0	0	1	EN1 bit (GCN2 register)
0	0	1	0	EN2 bit (GCN2 register)
0	0	1	1	EN3 bit (GCN2 register)

0	1	0	0	16-bit reload timer 0/2/4/6/8/10/12/14
0	1	0	1	16-bit reload timer 1/3/5/7/9/11/13/15
1	0	0	0	External trigger 0/4/8/12/16/20/24/28
1	0	0	1	External trigger 1/5/9/13/17/21/25/29
1	0	1	0	External trigger 2/6/10/14/18/22/26/30
1	0	1	1	External trigger 3/7/11/15/19/23/27/31
None of the above				Disabled (See “ 39.8. Caution (Page No.974) ”).)

- PPG0 to PPG15 as selected are activated when the edge specified by the Trigger Input Edge Selection bits (PCN.EGS[1:0]) are detected during the specified activation trigger.
- For detailed setting of each channel see chapter [39.7.7 What activation triggers are available and how are they selected? \(Page No.965\)](#)

39.4.5 GCN2: General Control Register 2

Generates PPG0-PPG3, PPG4-PPG7, PPG8-PPG11 and PPG12-PPG15 internal trigger levels using software.

- **GCN20 (PPG0-PPG3): Address 0103h (Access: Byte)**
- **GCN21 (PPG4-PPG7): Address 0107h (Access: Byte)**
- **GCN22 (PPG8-PPG11): Address 010Bh (Access: Byte)**
- **GCN23 (PPG12-PPG15): Address 0323h (Access: Byte)**
- **GCN24 (PPG16-PPG19): Address 0503h (Access: Byte)**
- **GCN25 (PPG20-PPG23): Address 0507h (Access: Byte)**
- **GCN26 (PPG24-PPG27): Address 050Bh (Access: Byte)**
- **GCN27 (PPG28-PPG31): Address 050Fh (Access: Byte)**

7	6	5	4	3	2	1	0	bit
–	–	–	–	EN3	EN2	EN1	EN0	
0	0	0	0	0	0	0	0	Initial value
R/W0	R/W0	R/W0	R/W0	R/W	R/W	R/W	R/W	Attribute

(See “Meaning of Bit Attribute Symbols (Page No.15)” for details of the attributes.)

- Bits 7-4: Undefined. Always write “0”. The read value is the value as written. (See “39.8. Caution (Page No.974)”.)
- Bit 3: EN3 trigger input
- Bit 2: EN2 trigger input
- Bit 1: EN1 trigger input
- Bit 0: EN0 trigger input

EN0, EN1, EN2, and EN3	Internal Triggers EN0, EN1, EN2, and EN3
0	Set the level to “L”.
1	Set the level to “H”.

- Set the levels of internal triggers EN0, EN1, EN2, and EN3.
- If any of the EN trigger inputs (EN0, EN1, EN2, EN3) is selected with the trigger specification bits (GCN10.TSEL0[3:0], GCN10.TSEL1[3:0], GCN10.TSEL2[3:0], and GCN10.TSEL3[3:0]) of PPG0, PPG1, PPG2, PPG3, then the selected EN serves as a PPG trigger input bit.
- If any of the EN trigger inputs (EN0, EN1, EN2, EN3) is selected with the trigger specification bits (GCN11.TSEL0[3:0], GCN11.TSEL1[3:0], GCN11.TSEL2[3:0], and GCN11.TSEL3[3:0]) of PPG4, PPG5, PPG6, PPG7, then the selected EN serves as a PPG trigger input bit.
- If any of the EN trigger inputs (EN0, EN1, EN2, EN3) is selected with the trigger specification bits (GCN12.TSEL0[3:0], GCN12.TSEL1[3:0], GCN12.TSEL2[3:0], and GCN12.TSEL3[3:0]) of PPG8, PPG9, PPG10, PPG11, then the selected EN serves as a PPG trigger input bit.
- If any of the EN trigger inputs (EN0, EN1, EN2, EN3) is selected with the trigger specification bits (GCN13.TSEL0[3:0], GCN13.TSEL1[3:0], GCN13.TSEL2[3:0], and GCN13.TSEL3[3:0]) of PPG12, PPG13, PPG14, of PPG15, then the selected EN serves as a PPG trigger input bit.
- If the state selected with the trigger input edge selection bit (EGS[1:0]) is generated by software using the trigger input bit (selected EN0, EN1, EN2, or EN3), the choice serves as an activation trigger to activate the PPG.

MB91460 Series**39.4.6 PTMR: PPG Timer Register**

Reads the counts of PPG0-PPG3, PPG4-PPG7, PPG8-PPG11 and PPG12-PPG15.

- **PTMR00 (PPG0): Address 0110h (Access: Half-word)**
- **PTMR01 (PPG1): Address 0118h (Access: Half-word)**
- **PTMR02 (PPG2): Address 0120h (Access: Half-word)**
- **PTMR03 (PPG3): Address 0128h (Access: Half-word)**
- **PTMR04 (PPG4): Address 0130h (Access: Half-word)**
- **PTMR05 (PPG5): Address 0138h (Access: Half-word)**
- **PTMR06 (PPG6): Address 0140h (Access: Half-word)**
- **PTMR07 (PPG7): Address 0148h (Access: Half-word)**
- **PTMR08 (PPG8): Address 0150h (Access: Half-word)**
- **PTMR09 (PPG9): Address 0158h (Access: Half-word)**
- **PTMR10 (PPG10): Address 0160h (Access: Half-word)**
- **PTMR11 (PPG11): Address 0168h (Access: Half-word)**
- **PTMR12 (PPG12): Address 0330h (Access: Half-word)**
- **PTMR13 (PPG13): Address 0338h (Access: Half-word)**
- **PTMR14 (PPG14): Address 0340h (Access: Half-word)**
- **PTMR15 (PPG15): Address 0348h (Access: Half-word)**
- **PTMR16 (PPG16): Address 0510h (Access: Half-word)**
- **PTMR17 (PPG17): Address 0518h (Access: Half-word)**
- **PTMR18 (PPG18): Address 0520h (Access: Half-word)**
- **PTMR19 (PPG19): Address 0528h (Access: Half-word)**
- **PTMR20 (PPG20): Address 0530h (Access: Half-word)**
- **PTMR21 (PPG21): Address 0538h (Access: Half-word)**
- **PTMR22 (PPG22): Address 0540h (Access: Half-word)**
- **PTMR23 (PPG23): Address 0548h (Access: Half-word)**
- **PTMR24 (PPG24): Address 0550h (Access: Half-word)**
- **PTMR25 (PPG25): Address 0558h (Access: Half-word)**
- **PTMR26 (PPG26): Address 0560h (Access: Half-word)**
- **PTMR27 (PPG27): Address 0568h (Access: Half-word)**
- **PTMR28 (PPG28): Address 0570h (Access: Half-word)**
- **PTMR29 (PPG29): Address 0578h (Access: Half-word)**
- **PTMR30 (PPG30): Address 0581h (Access: Half-word)**
- **PTMR31 (PPG31): Address 0588h (Access: Half-word)**

15	14	13	12	11	10	9	8	Bit
D15	D14	D13	D12	D11	D10	D9	D8	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

- The count of the 16-bit down counter can be read.
- Be sure to access the PPG Timer register PTMR in half words (16 bits).

- The register is not be read correctly if it is byte-accessed.

MB91460 Series

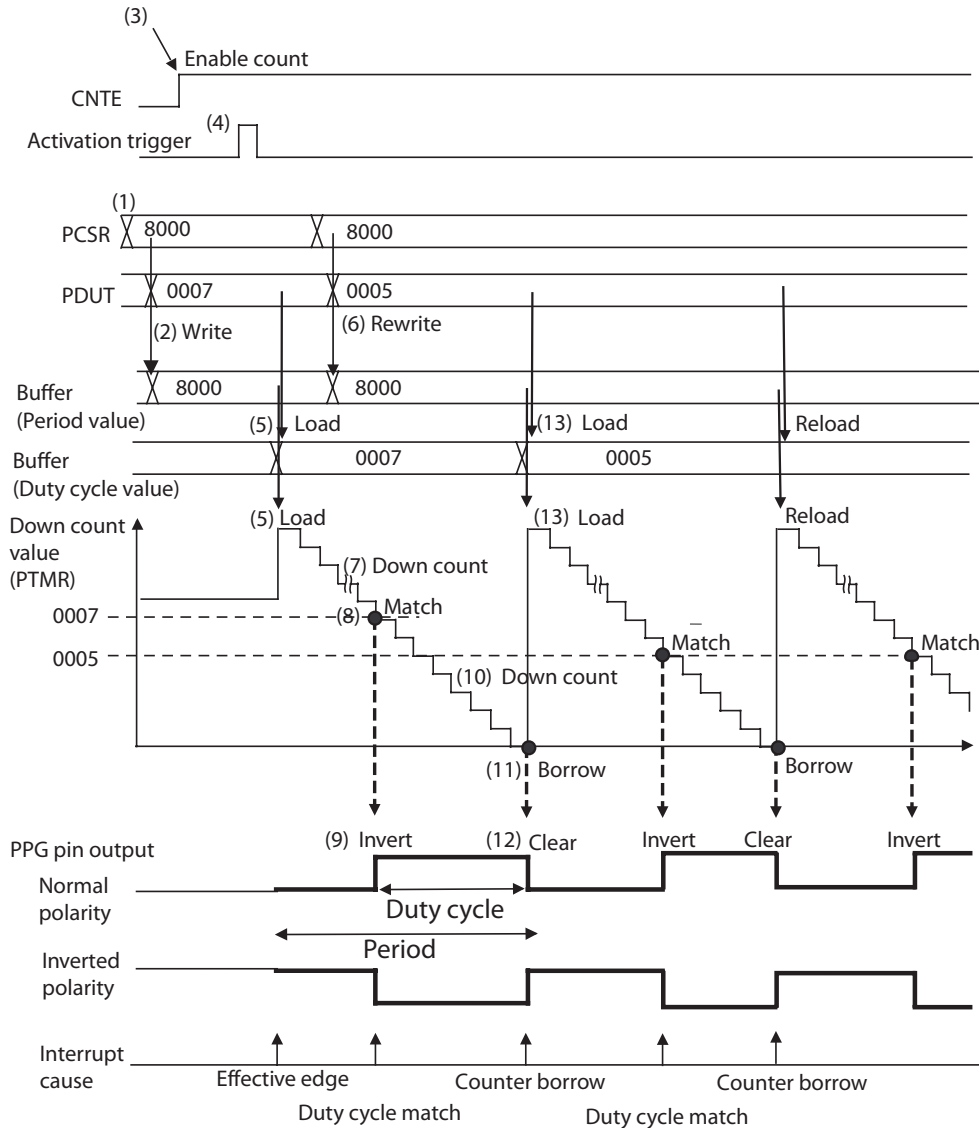
39.5. Operation

The MB91460 series features a maximum of 16 programmable pulse generators (PPGs), which provide programmable pulse output independently or jointly.

The individual modes of operation are described below.

39.5.1 PWM Operation

Figure 39.5-1 In PWM operation, variable duty cycle pulses are generated from the PPG pin



- (1) Write a period value.
- (2) Write a duty cycle value and transfer the period value to buffers.
- (3) Enable PPG operation.
- (4) Generate an activation trigger.
- (5) Load the period and duty cycle values.
- (6) Rewrite the duty cycle value and transfer the period value to buffers.
- (7) Counter counts down
- (8) The down counter equals the duty cycle value.

- (9) Inverses the PPG pin output level.
 - (10) Counter counts down
 - (11) Counter borrow (underflow)
 - (12) Clear the PPG pin output level (return to normal).
 - (13) Reload the period value.
 - (14) Reload the duty cycle value.
 - (15) Steps from (6) to (13) are iterated.
- (See “39.8. Caution (Page No.974)”.)

- Equation

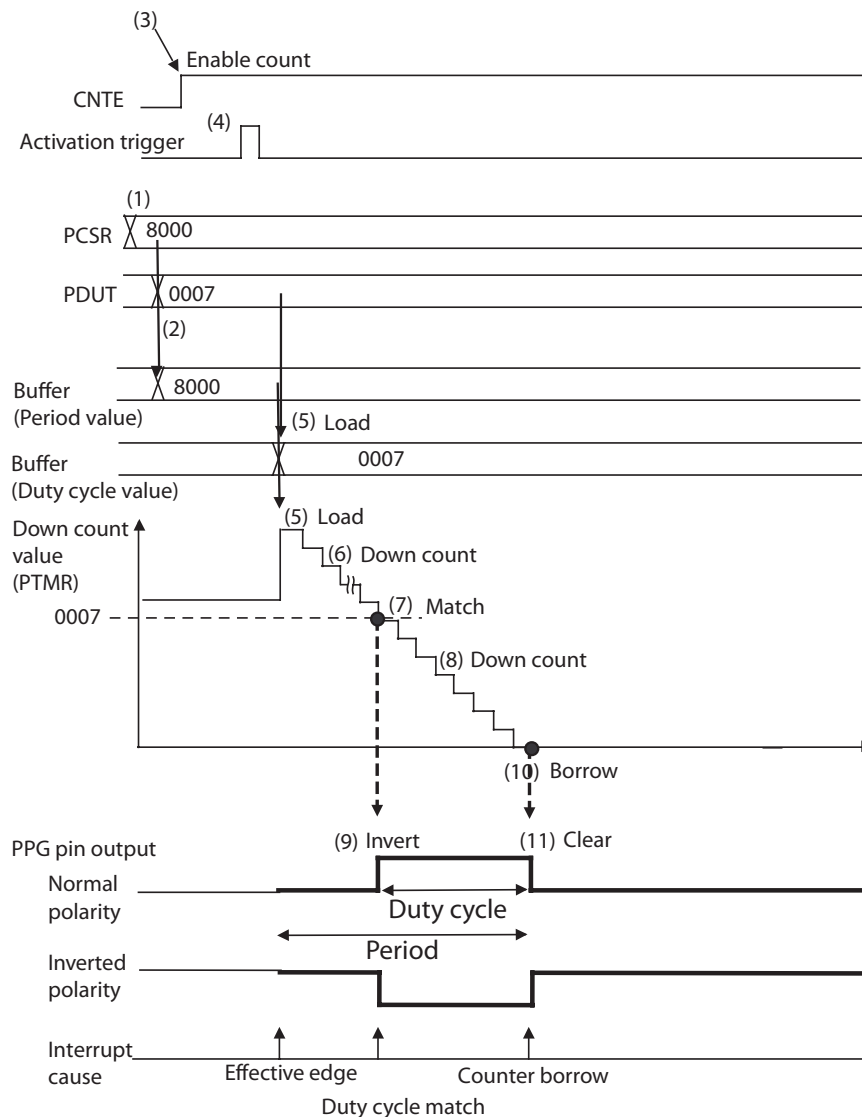
Period = {Period value (PCSR) + 1} x Count clock

Duty cycle = {Duty cycle value (PDUT) + 1} x Count clock

Width up to pulse output = {Period value (PCSR) – Duty cycle value (PDUT)} x Count clock

39.5.2 One-Shot Operation

In one-shot operation, one-shot pulses are generated from the PPG pin.

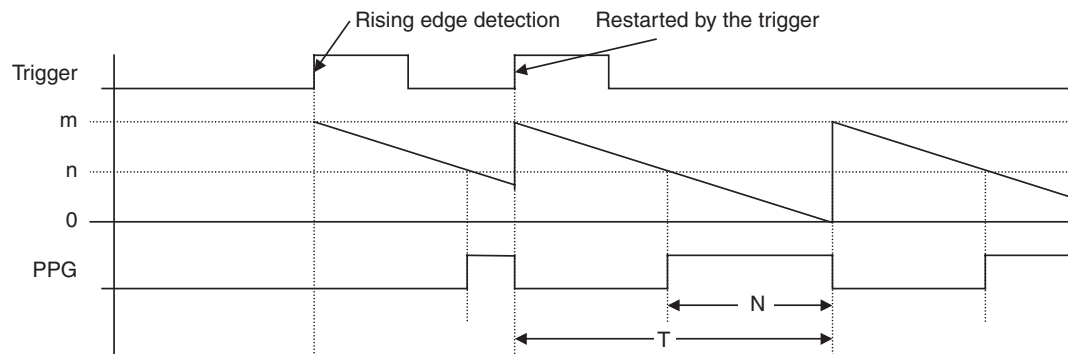


- (1) Write a period value.
- (2) Write a duty cycle value and transfer the period value to buffers.
- (3) Enable PPG operation.
- (4) Generate an activation trigger.
- (5) Load the period and duty cycle values.
- (6) Counter counts down
- (7) Down counter value and duty cycle value matches
- (8) Counter counts down
- (9) Inverse the PPG pin output level.
- (10) Counter borrow (underflow)
- (11) Clear the PPG pin output level (return to normal).
- (12) The operating sequence is now completed.
(See “[39.8. Caution \(Page No.974\)](#)”).)

39.5.3 Restart Operation

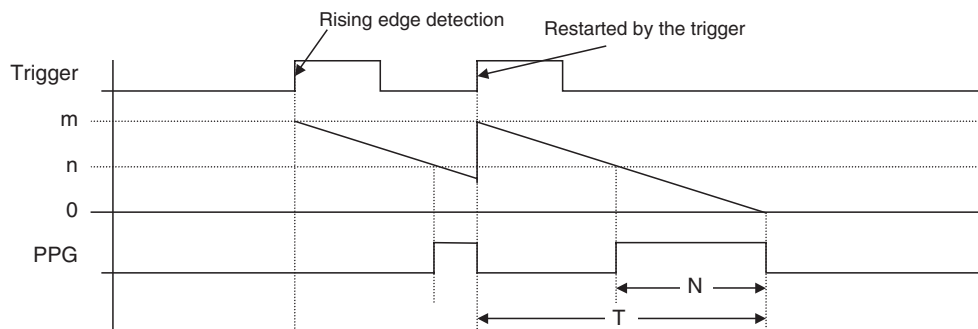
The restart operation is described below.

Figure 39.5-2 Restart available in PWM operation



N = duty cycle, T = period

Figure 39.5-3 Restart available in one-shot operation



If a restart is not available, the second and subsequent triggers have no effect in both PWM and one-shot operations.

(The second and subsequent triggers following a shutdown of the down counter are functional.)

MB91460 Series**39.6. Setting****Table 39.6-1 Settings Needed to Start the PPG**

Setting	Setting Registers	Setting Procedure*
Period and duty cycle value settings	PPG period settings (PCSR00-PCSR31) PPG duty cycle settings (PDUT00-PDUT31)	7.1
Enable PPG operation.	PPG control status (PCN00-PCN31)	7.2
Operation mode selection (PWM/one-shot)		7.3
Enable restart.		7.4
Count clock selection		7.5
PPG output mask selection		7.6
Trigger selection Software Internal trigger External trigger	General Control 1 (GCN10, GCN11, GCN12, GCN13, GCN14, GCN15, GCN16, GCN17)	7.7
Output polarity specification		7.8
PPG pin output setting	Port functions (PFR16, PFR17)	7.9
Trigger generation (software trigger) (Reload timer) (GCN2.EN bit)	PPG Control Status (PCN00-PCN31)	7.10
	See “ Chapter 38 Reload Timer (RLT) (Page No.921)”.	
	General Control 2 (GCN20, GCN21, GCN22, GCN23, GCN24, GCN25, GCN26, GCN27)	

* For refer to the section indicated by the number.

Table 39.6-2 Settings Needed to Stop the PPG

Setting	Setting Registers	Setting Procedure*
PPG stop bit setting	PPG control status (PCN00-PCN31)	7.11

*For the setting procedure, refer to the section indicated by the number.

Table 39.6-3 Settings Needed to Clamp the Output Level

Setting	Setting Registers	Setting Procedure*
Output polarity specification	PPG control status (PCN00-PCN31)	7.8
PPG output mask selection		7.6
Period value = Duty cycle value setting	PPG duty cycle settings (PDUT00-PDUT31)	7.6

*For the setting procedure, refer to the section indicated by the number.

Table 39.6-4 Settings Needed to Implement PPG Interrupts

Setting	Setting Registers	Setting Procedure*
PPG interrupt vector, PPG interrupt level setting	See “ Chapter 24 Interrupt Control (Page No.429)”.	7.12

Table 39.6-4 Settings Needed to Implement PPG Interrupts

PPG interrupt cause selection (Generate an activation trigger, borrow, and duty cycle match)	PPG control status (PCN00-PCN31)	7.13
PPG interrupt setting Clear interrupts. Enable interrupt requests.		7.14

*:For the setting procedure, refer to the section indicated by the number.

39.7. Q & A

39.7.1 How to set (rewrite) a period and a duty cycle

Period and duty cycle value settings

- Set each period value in PPG Period Setting Register PCSR.
- Set each duty cycle value in PPG Duty cycle Setting Register PDUT.
- The PPG Period Setting and the PPG Duty cycle Setting registers each have a buffer to allow the user to ignore the write timing.

- Equation

PCSR register value = {Period/Count clock} – 1

PDUT register value = {"H" width (duty cycle)^{*}/Count clock} – 1

*: Normal polarity (OSEL= 0)

- Allowed range

PCSR register value = PCSR register value - FFFFh (65535)

PDUT register value = 0 - PCSR register value

Note: Be sure to set the duty cycle after the setting of the period. (See “39.8. Caution (Page No.974)”.)

39.7.2 How to enable or disable PPG operations

Enabling the PPG operation

Use the PPG operation enable bit (PCN.CNTE).

Control	PPG Operation Enable Bit (CNTE)
To stop a PPG operation	Set “0”.
To enable a PPG operation	Set “1”.

Enable PPG operation before starting the PPG.

(See “39.8. Caution (Page No.974)”.)

39.7.3 How to set the PPG operation mode (PWM operation/one-shot operation)?

Operation mode selection

Use the mode selection bit (PCN.MDSE).

Operation Mode	Mode Selection Bit (MDSE)
To implement a PWM operation	Set “0”.
To implement a one-shot operation	Set “1”.

(See “39.8. Caution (Page No.974)”.)

39.7.4 How to get it restarted

Enable restart.

A restart of a PPG can be enabled while the PPG is in operation.

Use the Enable Restart bit (PCN.RTRG) to set.

(See “39.8. Caution (Page No.974)”.)

39.7.5 What count clocks are available and how are they selected?

Count clock selection

The count clock is selectable out of the four choices listed below.

Use the count clock selection bit (PCN.CKS[1:0]).

Count Clock	Count Clock Selection Bit		(Example) CLKP = 32MHz	
	CKS1	CKS0	Count Clock	Period (1 - FFFFh)
CLKP	0	0	32MHz	62.5ns - 2.048ms
CLKP/4	0	1	8MHz	250ns - 8.192ms
CLKP/16	1	0	2MHz	1μs - 32.76ms
CLKP/64	1	1	500kHz	4μs - 131.0ms

(See “39.8. Caution (Page No.974)”.)

39.7.6 How to clamp the PPG pin output level

PPG output mask selection

The level of PPG pin output can be clamped.

Use the PPG Output Mask Selection bit (PCN.PGMS) and the duty cycle value (PDUT) to set.

PPG Pin Output	PPG Output Polarity Specification Bit (OSEL)	Setting Procedure
To clamp the “L” level under normal polarity	When “0”	Set the PPG Output Mask Selection bit (PGMS) to “1”.
To clamp the “H” level under normal polarity	When “0”	Period value (PCSR) = Set a duty cycle value (PDUT).
To clamp the “H” level under inverted polarity	When “1”	Set the PPG Output Mask Selection bit (PGMS) to “1”.
To clamp the “L” level under inverted polarity	When “1”	Period value (PCSR) = Set a duty cycle value (PDUT).

Figure 39.7-1 PPG pin output can be set to all “L”. (when OSEL=“0”)

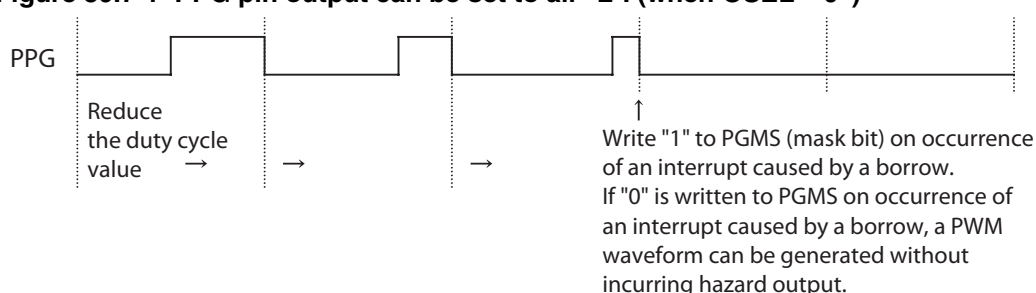
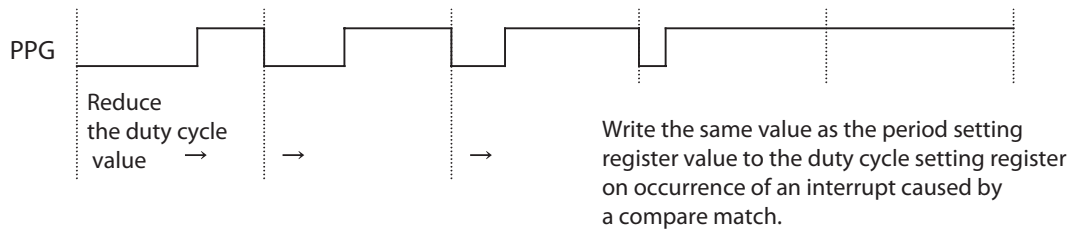


Figure 39.7-2 PPG pin output can be set to all “H”. (when OSEL=“0”)

PPG output will also equal all “H” if “0” is set in both the PPG Period Setting Register (PCSR) and PPG Duty cycle Setting Register (PDUT). (when OSEL=“0”)

39.7.7 What activation triggers are available and how are they selected?

- Trigger selection
 - Activation triggers are broadly grouped into software triggers, internal triggers and external triggers.
 - Software triggers work at all times.
 - Internal and external trigger availability depends on each device specification.

An trigger is set using the trigger specification bits (GCN1.TSEL0[3:0]), (GCN1.TSEL1[3:0]), (GCN1.TSEL2[3:0]), and (GCN1.TSEL3[3:0]).

Triggers are selectable for PPG0, PPG1, PPG2 and PPG3 independently.

Internal Trigger	PPG0	PPG1	PPG2	PPG3
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN20 register	Set “0000”	Set “0000”	Set “0000”	Set “0000”
To select the EN1 bit of the GCN20 register	Set “0001”	Set “0001”	Set “0001”	Set “0001”
To select the EN2 bit of the GCN20 register	Set “0010”	Set “0010”	Set “0010”	Set “0010”
To select the EN3 bit of the GCN20 register	Set “0011”	Set “0011”	Set “0011”	Set “0011”
To select reload timer 0	Set “0100”	Set “0100”	Set “0100”	Set “0100”
To select reload timer 1	Set “0101”	Set “0101”	Set “0101”	Set “0101”

External Trigger	PPG0	PPG1	PPG2	PPG3
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_0 (TTG0)	Set “1000”	Set “1000”	Set “1000”	Set “1000”
To select the external trigger on GP14_1 (TTG1)	Set “1001”	Set “1001”	Set “1001”	Set “1001”
To select the external trigger on GP14_2 (TTG2)	Set “1010”	Set “1010”	Set “1010”	Set “1010”
To select the external trigger on GP14_3 (TTG3)	Set “1011”	Set “1011”	Set “1011”	Set “1011”

Triggers are selectable for PPG4, PPG5, PPG6 and PPG7 independently.

Internal Trigger	PPG4	PPG5	PPG6	PPG7
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN21 register	Set “0000”	Set “0000”	Set “0000”	Set “0000”
To select the EN1 bit of the GCN21 register	Set “0001”	Set “0001”	Set “0001”	Set “0001”

To select the EN2 bit of the GCN21 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN21 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 2	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 3	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG4	PPG5	PPG6	PPG7
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_4 (TTG4)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_5 (TTG5)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_6 (TTG6)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_7 (TTG7)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG8, PPG9, PPG10 and PPG11 independently.

Internal Trigger	PPG8	PPG9	PPG10	PPG11
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN22 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN22 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN22 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN22 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 4	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 5	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG8	PPG9	PPG10	PPG11
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_0 (TTG0)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_1 (TTG1)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_2 (TTG2)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_3 (TTG3)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG12, PPG13, PPG14 and PPG15 independently.

Internal Trigger	PPG12	PPG13	PPG14	PPG15
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN23 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN23 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN23 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN23 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"

MB91460 Series

To select reload timer 6	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 7	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG12	PPG13	PPG14	PPG15
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_4 (TTG4)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_5 (TTG5)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_6 (TTG6)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_7 (TTG7)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG16, PPG17, PPG18 and PPG19 independently.

Internal Trigger	PPG16	PPG17	PPG18	PPG19
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN24 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN24 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN24 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN24 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 8	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 9	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG16	PPG17	PPG18	PPG19
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_0 (TTG16)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_1 (TTG17)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_2 (TTG18)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_3 (TTG19)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG20, PPG21, PPG22 and PPG23 independently.

Internal Trigger	PPG20	PPG21	PPG22	PPG23
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN25 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN25 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN25 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN25 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 10	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 11	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG20	PPG21	PPG22	PPG23
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_4 (TTG20)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_5 (TTG21)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_6 (TTG22)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_7 (TTG23)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG24, PPG25, PPG26 and PPG27 independently.

Internal Trigger	PPG24	PPG25	PPG26	PPG27
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN26 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN26 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN26 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN26 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 12	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 13	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG24	PPG25	PPG26	PPG27
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_0 (TTG24)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_1 (TTG25)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_2 (TTG26)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_3 (TTG27)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

Triggers are selectable for PPG28, PPG29, PPG30 and PPG31 independently.

Internal Trigger	PPG28	PPG29	PPG30	PPG31
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN27 register	Set "0000"	Set "0000"	Set "0000"	Set "0000"
To select the EN1 bit of the GCN27 register	Set "0001"	Set "0001"	Set "0001"	Set "0001"
To select the EN2 bit of the GCN27 register	Set "0010"	Set "0010"	Set "0010"	Set "0010"
To select the EN3 bit of the GCN27 register	Set "0011"	Set "0011"	Set "0011"	Set "0011"
To select reload timer 14	Set "0100"	Set "0100"	Set "0100"	Set "0100"
To select reload timer 15	Set "0101"	Set "0101"	Set "0101"	Set "0101"

External Trigger	PPG28	PPG29	PPG30	PPG31
	External Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the external trigger on GP14_4 (TTG28)	Set "1000"	Set "1000"	Set "1000"	Set "1000"
To select the external trigger on GP14_5 (TTG29)	Set "1001"	Set "1001"	Set "1001"	Set "1001"
To select the external trigger on GP14_6 (TTG30)	Set "1010"	Set "1010"	Set "1010"	Set "1010"
To select the external trigger on GP14_7 (TTG31)	Set "1011"	Set "1011"	Set "1011"	Set "1011"

The same trigger can be specified for a group of PPGs to activate all these PPGs simultaneously. (See "39.8. Caution (Page No.974)".)

- Trigger edge selection

Trigger edges are set using trigger input edge selection bits (PCN.EG[1:0]).

Trigger Edge Selection	Trigger Input Edge Selection Bits (EG1-EG0)
When not detected (software trigger only)	Set "00".
"L" -> "H" Trigger generated on the rising edge	Set "01".
"H" -> "L" Trigger generated on the falling edge	Set "10".
Trigger generated on both edges	Set "11".

(See "39.8. Caution (Page No.974)".)

39.7.8 How to invert the output polarity?

Output polarity specification

The polarity in the normal state can be specified as follows:

Use the PPG Output Polarity Specification bit (PCN.OSEL) to set.

("Normal state" means the state in which pulse output is not executed.)

Output Level in Normal State	PPG Output Polarity Specification Bit (OSEL)
To enable "L" level output (normal polarity)	Set "0".
To enable "H" level output (inverted polarity)	Set "1".

(See "39.8. Caution (Page No.974)".)

39.7.9 How to program a pin as a PPG output pin?

-> PPG pin output setting

Software programming allows ports to be switched to PPG pin output.

Pin	Control Bit Location	
PPG0 pin	Port Function register PFR17.0 = '1'	PPG0 Output specification bit (PPG0)
PPG1 pin	Port Function register PFR17.1 = '1'	PPG1 Output specification bit (PPG1)
PPG2 pin	Port Function register PFR17.2 = '1'	PPG2 Output specification bit (PPG2)
PPG3 pin	Port Function register PFR17.3 = '1'	PPG3 Output specification bit (PPG3)
PPG4 pin	Port Function register PFR17.4 = '1'	PPG4 Output specification bit (PPG4)
PPG5 pin	Port Function register PFR17.5 = '1'	PPG5 Output specification bit (PPG5)
PPG6 pin	Port Function register PFR17.6 = '1'	PPG6 Output specification bit (PPG6)
PPG7 pin	Port Function register PFR17.7 = '1'	PPG7 Output specification bit (PPG7)
PPG8 pin	Port Function register PFR16.0 = '1'	PPG8 Output specification bit (PPG8)
PPG9 pin	Port Function register PFR16.1 = '1'	PPG9 Output specification bit (PPG9)
PPG10 pin	Port Function register PFR16.2 = '1'	PPG10 Output specification bit (PPG10)
PPG11 pin	Port Function register PFR16.3 = '1'	PPG11 Output specification bit (PPG11)
PPG12 pin	Port Function register PFR16.4 = '1'	PPG12 Output specification bit (PPG12)
PPG13 pin	Port Function register PFR16.5 = '1'	PPG13 Output specification bit (PPG13)
PPG14 pin	Port Function register PFR16.6 = '1'	PPG14 Output specification bit (PPG14)
PPG15 pin	Port Function register PFR16.7 = '1'	PPG15 Output specification bit (PPG15)
PPG16 pin	Port Function register (E)PFR30.0 = '1'	PPG16 Output specification bit (PPG16)
PPG17 pin	Port Function register (E)PFR30.1 = '1'	PPG17 Output specification bit (PPG17)
PPG18 pin	Port Function register (E)PFR30.2 = '1'	PPG18 Output specification bit (PPG18)
PPG19 pin	Port Function register (E)PFR30.3 = '1'	PPG19 Output specification bit (PPG19)
PPG20 pin	Port Function register (E)PFR30.4 = '1'	PPG20 Output specification bit (PPG20)
PPG21 pin	Port Function register (E)PFR30.5 = '1'	PPG21 Output specification bit (PPG21)
PPG22 pin	Port Function register (E)PFR30.6 = '1'	PPG22 Output specification bit (PPG22)
PPG23 pin	Port Function register (E)PFR30.7 = '1'	PPG23 Output specification bit (PPG23)
PPG24 pin	Port Function register (E)PFR35.3 = '1'	PPG0 Output specification bit (PPG24)

PPG25 pin	Port Function register (E)PFR35.7 = '1'	PPG1 Output specification bit (PPG25)
PPG26 pin	Port Function register (E)PFR34.3 = '1'	PPG2 Output specification bit (PPG26)
PPG27 pin	Port Function register (E)PFR34.7 = '1'	PPG3 Output specification bit (PPG27)
PPG28 pin	Port Function register (E)PFR33.3 = '1'	PPG4 Output specification bit (PPG28)
PPG29 pin	Port Function register (E)PFR33.7 = '1'	PPG5 Output specification bit (PPG29)
PPG30 pin	Port Function register (E)PFR33.3 = '1'	PPG6 Output specification bit (PPG30)
PPG31 pin	Port Function register (E)PFR33.7 = '1'	PPG7 Output specification bit (PPG31)

39.7.10 How to generate an activation trigger?

Generating a trigger

Methods of generating an activation trigger are described below.

- Activating a software trigger
Use the Software Trigger bit (PCN.STGR) to set.
Write "1" to the Software Trigger bit (STGR) to generate an activation trigger.
Always functional, regardless of the internal trigger.
- Activating PPGs with reload timers
The reload timers need to be set up and activated. For more information, see "[Chapter 38 Reload Timer \(RLT\) \(Page No.921\)](#)".
An activation trigger is generated when the edge specified by the reload timer output signal is generated with the reload timer underflow.
- Activating PPGs with external triggers
An activation trigger is generated when the edge specified on the specified pin appears. There is no need for configuration of the port registers; the trigger is always connected to the PPGs.
- Activating a PPG with the EN trigger input bits (GCN2.EN0) - (GCN2.EN3)
An activation trigger can be generated by rewriting the level of the EN trigger input bits (GCN2.EN0) - (GCN2.EN3).

Edge	Software-Based Setting Procedure (EN0, EN1, EN2, EN3)
Rising edge	First, set the EN bit to "0", then the EN bit to "1".
Falling edge	First, set the EN bit to "1", then to "0".

- Activating multiple PPGs concurrently
The same trigger (trigger input bit) can be specified with the PPG trigger specification bits to activate all the PPGs simultaneously when the trigger is generated.
- Even if an activation trigger is generated before the operation of a PPG is enabled, that PPG would not be activated. Be sure to enable the operation of a PPG before generating a trigger to activate it. (See "[39.7.2 How to enable or disable PPG operations \(Page No.963\)](#)".)

39.7.11 How to stop a PPG operation?

PPG stop bit setting (See "[39.7.2 How to enable or disable PPG operations \(Page No.963\)](#)".)

39.7.12 What interrupt registers are used?

PPG interrupt vector, PPG interrupt level setting

The table below summarizes the relationships among the PPG number, interrupt level and interrupt vector.

For more information about the interrupt levels and interrupt vectors, see "[Chapter 24 Interrupt Control \(Page No.429\)](#)".

	Interrupt Vector (Default)	Interrupt Level Setting Bit (ICR[4:0])
--	----------------------------	--

PPG0 PPG16	#112 Address: 0FFE3Ch	Interrupt Level register (ICR48) Address: 0470h
PPG1 PPG17	#113 Address: 0FFE38h	
PPG2 PPG18	#114 Address: 0FFE34h	Interrupt Level register (ICR49) Address: 0471h
PPG3 PPG19	#115 Address: 0FFE30h	
PPG4 PPG20	#116 Address: 0FFE2Ch	Interrupt Level register (ICR50) Address: 0472h
PPG5 PPG21	#117 Address: 0FFE28h	
PPG6 PPG22	#118 Address: 0FFE24h	Interrupt Level register (ICR51) Address: 0473h
PPG7 PPG23	#119 Address: 0FFE20h	
PPG8 PPG24	#120 Address: 0FFE1Ch	Interrupt Level register (ICR52) Address: 0474h
PPG9 PPG25	#121 Address: 0FFE18h	
PPG10 PPG26	#122 Address: 0FFE14h	Interrupt Level register (ICR53) Address: 0475h
PPG11 PPG27	#123 Address: 0FFE10h	
PPG12 PPG28	#124 Address: 0FFE0Ch	Interrupt Level register (ICR54) Address: 0476h
PPG13 PPG29	#125 Address: 0FFE08h	
PPG14 PPG30	#126 Address: 0FFE04h	Interrupt Level register (ICR55) Address: 0477h
PPG15 PPG31	#127 Address: 0FFE00h	

The Interrupt flag (PCN.IRQF) does not clear itself automatically. Use software to clear it before returning from the interrupt handler. (Write "0" to the IRQF bit.)

39.7.13 What interrupts are available and how are they selected?

Interrupt cause selection

Four kinds of interrupts are selectable as follows:

Use the Interrupt Cause Setting bit (PCN.IRS[1:0]) to set.

Interrupt Cause	Interrupt Cause Setting Bit (IRS[1:0])
Software trigger or Internal trigger generation (PPG0-PPG15)	Set "00".
Down counter borrow (period match)	Set "01".
Duty cycle match	Set "10".
Down counter borrow (period match) or Duty cycle match	Set "11".

39.7.14 How to enable, disable and clear interrupts?

Interrupt Request Enable bit, Interrupt flag

Use the interrupt request enable bit (PCN.IREN) to enable interrupts.

	Interrupt Request Enable Bit (IREN)
To disable interrupt requests	Set "0".
To enable interrupt requests	Set "1".

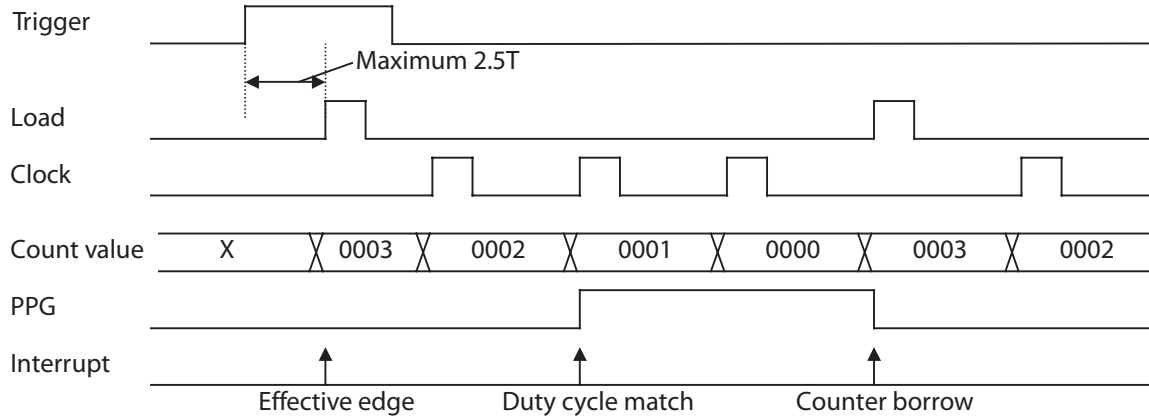
Use the interrupt flag (PCN.IRQF) to clear interrupts.

	Interrupt flag (IRQF)
To clear interrupts	Write "0".

(See "39.8. Caution (Page No.974)".)

39.8. Caution

- If an interrupt occurs (Interrupt flag PCN.IRQF is set to “1”) and the Interrupt flag is set to “0” at the same time, the setting of the Interrupt flag to “1” has higher priority.
- The first load comes with a maximum delay of $2.5T$ after the activation trigger. (T: Count clock)
If the down counter is loaded and counts at the same time, the load operation overrides.



- Be sure to write duty cycle value PDUT after period value PCSR has been initialized and rewritten. (Always write in the order of (1)PCSR and (2)PDUT.)
Only the PDUT can be written for rewriting the duty cycle.
- Set the duty cycle value PDUT smaller than the period value PCSR. If any larger value has been set, disable the operation of the PPG before replacing the duty cycle with a smaller value.
- Always access PPG Period Setting registers PCSR and PPG Duty cycle Setting registers in a half-word (16-bit) format. If these registers are byte-accessed, no values are written to their upper and lower bit positions.
- To activate a PPG, it is necessary to set the Timer Operation Enable bits (PCN.CNTE) to “1” before or concurrently with the activation to enable the PPG operation.
- The values of mode (MDSE), restart enable (RTRG), count clock (CKS[1:0]), trigger input edge (EGS[1:0]), interrupt cause (IRS), internal trigger (TSEL) and output polarity specification (OSEL) may not be changed while the PPG is operating.
If any of these values has been changed while the PPG was operating, disable the operation of the PPG before reloading the register.
- Whenever writing a value to GCN2, be sure to write “0” to any undefined part of the upper 4 bits.
If “1” is written, disable the operation of the PPG before reloading the register.
- If any value outside the specified range (0110, 0111, 1100 - 1111) is set in Activation Trigger Specification bits (TSEL0[3:0]), (TSEL1[3:0]), (TSEL2[3:0]), (TSEL3[3:0]) has been set, disable the operation of the PPG and then write the specified value to let the register return to normal.
- If the Timer Operation Enable bit (PCN.CNTE) is set to “0” to disable PPGn while it is operating, the PPG stops and sets its output value to the initial value (“1” if (OSEL “1”) else “0”). The PPG timer is latched until the operation of the timer is enabled by setting the Timer Operation enable bit (PCN.CNTE) to “1”.
- If (PCN.CNTE) is set to “1” to enable the PPG, it restarts (PPG timer is set to initial value).

Chapter 40 Pulse Frequency Modulator (PFM)

This chapter provides an overview of the 16-bit pulse frequency modulator, describes the register structure/functions, and describes the operation of the 16-bit pulse frequency modulator.

40.1. PFM Overview

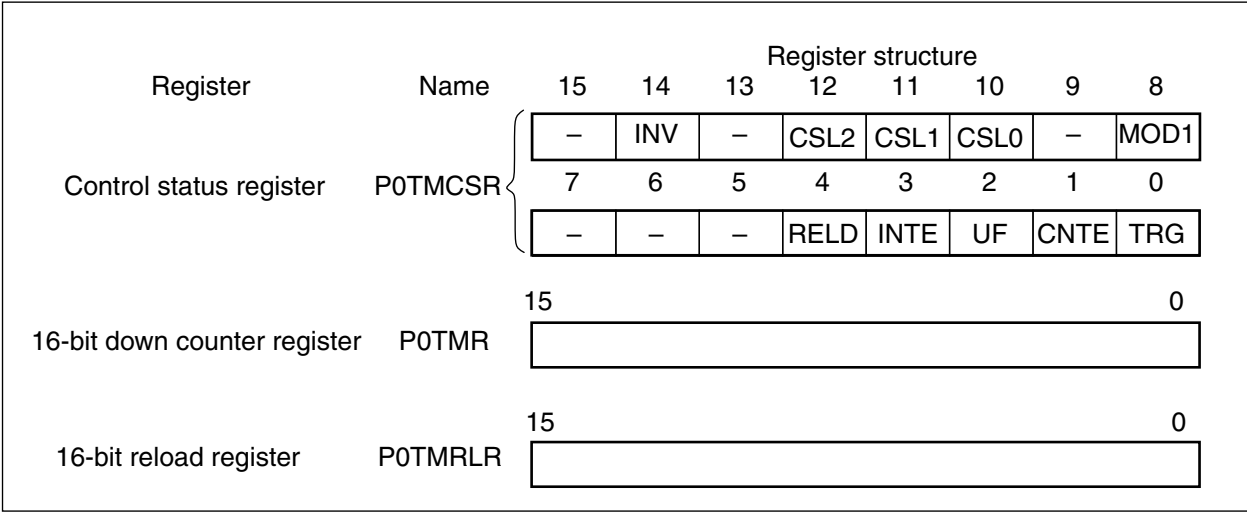
The 16-bit pulse frequency modulator consists of two 16-bit down-counters, two 16-bit reload registers, prescalers for generating the internal count clocks and control/status registers. The PFM is used to generate pulses of a short duration in a long period. This is an alternative to using PWM signals in some applications.

■ Functions

- Two independent programmable 16-bit down-counters generating low and high pulses.
- The input clock (count clock) can be selected from prescaled internal clocks (the peripheral clock (CLKP) divided by 2/8/32/64/128) separately for each counter.
- The mark level and output waveform can be inverted.

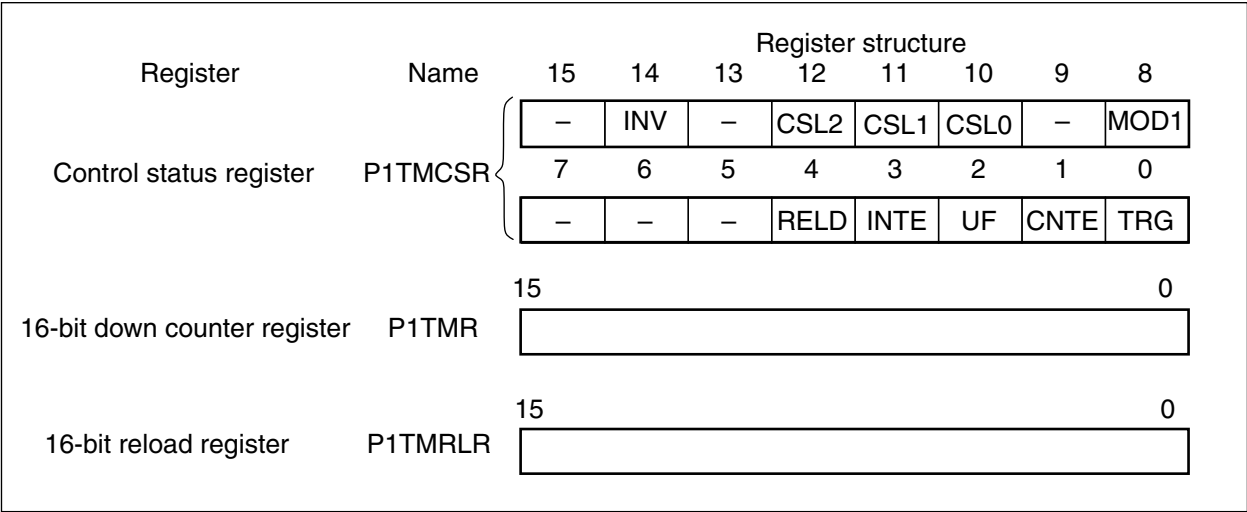
■ 16-bit Reload Counter 0 Register Configuration

Figure 40.1-1 16-bit Reload Counter 0 Register Configuration

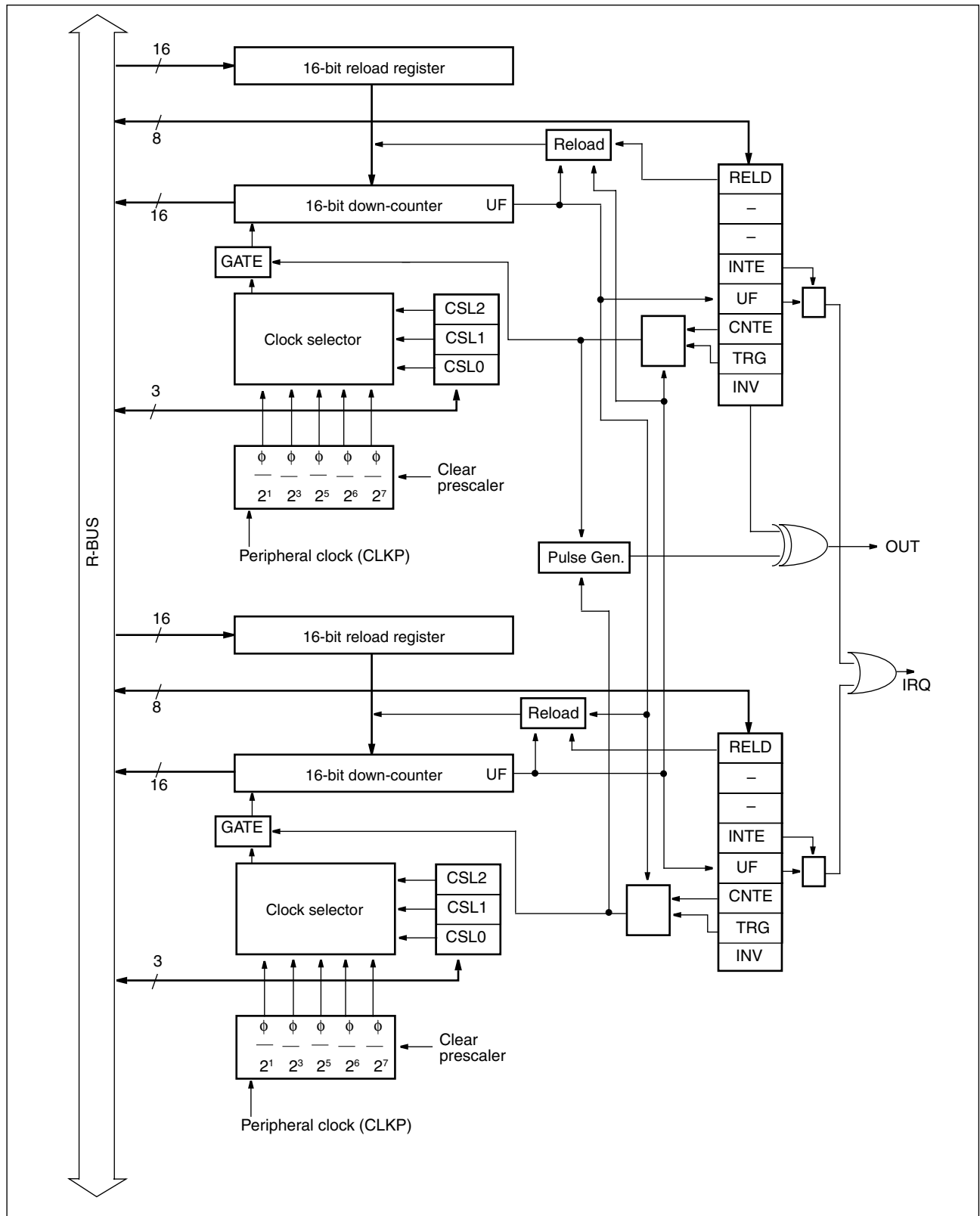


■ 16-bit Reload Counter 1 Register Configuration

Figure 40.1-2 16-bit Reload Counter 1 Register Configuration



0FFE48H 32-bit (interrupt vector #109)

MB91460 Series**■ Block Diagram of the 16-Bit Pulse Frequency Modulator****Figure 40.1-3 Block Diagram of the 16-bit Pulse Frequency Modulator**

40.2. Reload Counter Registers

This section describes the 16-bit pulse frequency modulator registers listed below.

Control status register (P0TMCSR, P1TMCSR)

16-bit counter register (P0TMR, P1TMR)

16-bit reload register (P0TMRLR, P1TMRLR)

40.2.1 Control Status Register (P0TMCSR, P1TMCSR)

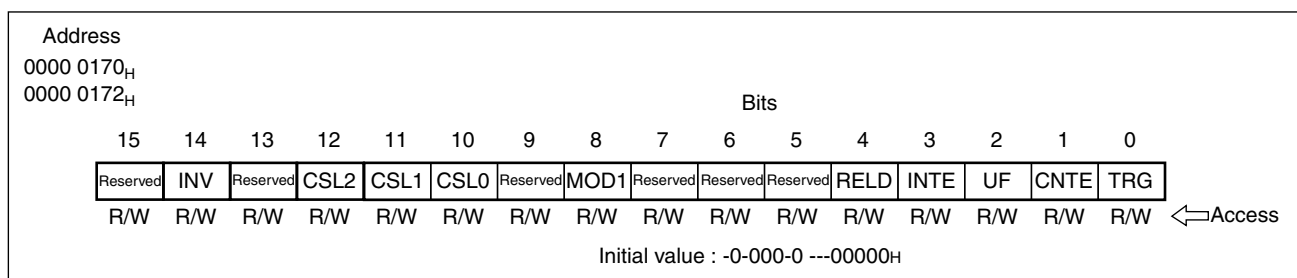
Controls the operation mode, shows the status of the reload counter and interrupts for the 16-bit reload counter.

Only change the value of bits other than UF and TRG when CNTE = "0".

The bits can be written simultaneously.

● P0TMCSR, P1TMCSR structure

Figure 40.2-1 Structure of the Control Status Register



● Functions of the P0TMCSR, P1TMCSR bits

[Bit 15] Reserved

Always set to "0".

[Bit 14] INV (INVersion)

The output signal inversion bit.

"0" is default level (counter 0 high level, counter 1 low level).

"1" inverts the output signal (counter 0 low level, counter 1 high level).

Remark: Writing INV of P0TMCSR or INV of P1TMCSR or both has the same effect.

[Bit 13] Reserved

Always set to "0".

[Bits 12, 10] CSL2, CSL1, CSL0 (Count clock SeLect)

The count clock select bits.

Table 40.2-1 lists the clock source selections.

Table 40.2-1 CSL Bit Clock Source Settings

CSL2	CSL1	CSL0	Clock source (Peripheral clock (CLKP))
0	0	0	CLKP / 2 ¹
0	0	1	CLKP / 2 ³
0	1	0	CLKP / 2 ⁵
0	1	1	reserved
1	0	0	Clock disabled
1	0	1	CLKP / 2 ⁶
1	1	0	CLKP / 2 ⁷
1	1	1	Clock disabled

[Bits 9] Reserved

Always set to "0".

[Bits 8] MOD1

MOD1	
0	Setting prohibited
1	Necessary for PFM operation

[Bits 7 to 5] Reserved

Always set to "010".

[Bit 4] RELD

This bit enables reload operations.

When RELD is "1", the counter operates in reload mode. In this mode, the counter loads the reload register contents into the counter and continues counting whenever an underflow occurs (when the counter value changes from 0000_H to FFFF_H).

When RELD is "0", the count operation stops when an underflow occurs due to the counter value changing from 0000_H to FFFF_H.

Note: For PFM operation it is necessary to set the RELD bits to "0".

[Bit 3] INTE

The interrupt request enable bit.

When INTE is "1", an interrupt request is generated when the UF bit changes to "1".

When INTE is "0", no interrupt requests are generated.

[Bit 2] UF

The counter interrupt flag.

UF is set to "1" when an underflow occurs (when the counter value changes from 0000_H to FFFF_H).

Writing "0" clears the flag. Writing "1" has no effect. Read as "1" by read-modify-write instructions.

[Bit 1] CNTE

The counter count enable bit.

Writing "1" sets the counter to wait for a trigger.

Writing "0" stops count operation.

[Bit 0] TRG

Software trigger bit.

Writing "1" to TRG applies a software trigger, causing the counter to load the reload register contents to the counter and start counting.

Writing "0" has no effect. Reading always returns "0".

Applying a trigger using this register is only valid when CNTE = "1". Writing "1" has no effect if CNTE = "0".

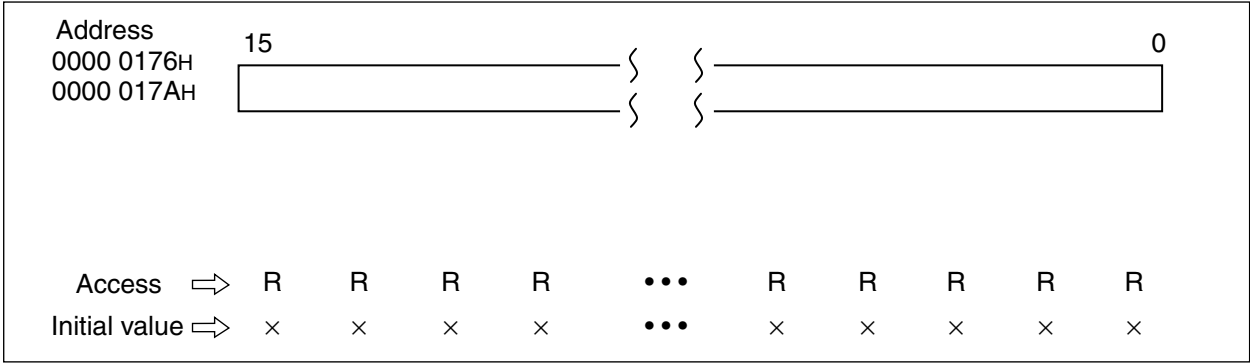
MB91460 Series

40.2.2 16-bit Counter Register (P0TMR, P1TMR)

Reading this register reads the count value of the 16-bit down counter.
The initial value is indeterminate.
Always read this register using 16-bit data transfer instructions.

- P0TMR, P1TMR structure

Figure 40.2-2 Structure of the 16-bit Counter Register

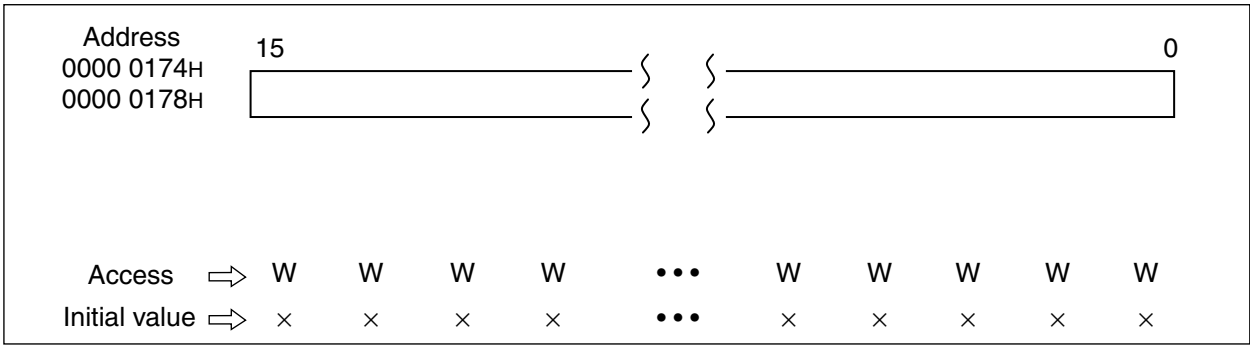


40.2.3 16-bit Reload Register (P0TMRLR, P1TMRLR)

The 16-bit reload register stores the initial count value.
The initial value is indeterminate.
Always write to this register using 16-bit data transfer instructions.

- P0TMRLR, P1TMRLR structure

Figure 40.2-3 Structure of the 16-bit Reload Register



40.3. Reload Counter Operation

This section describes the operations of the 16-bit reload counter: Internal clock operation and Underflow operation

40.3.1 Internal Clock Operation

The Peripheral Clock (CLKP) divided by 2, 8, 32, 64 or 128 can be selected as the clock source when operating the counter from an internal clock.

Writing "1" to both the CNTE and TRG bits in the control status register enables and starts counting simultaneously.

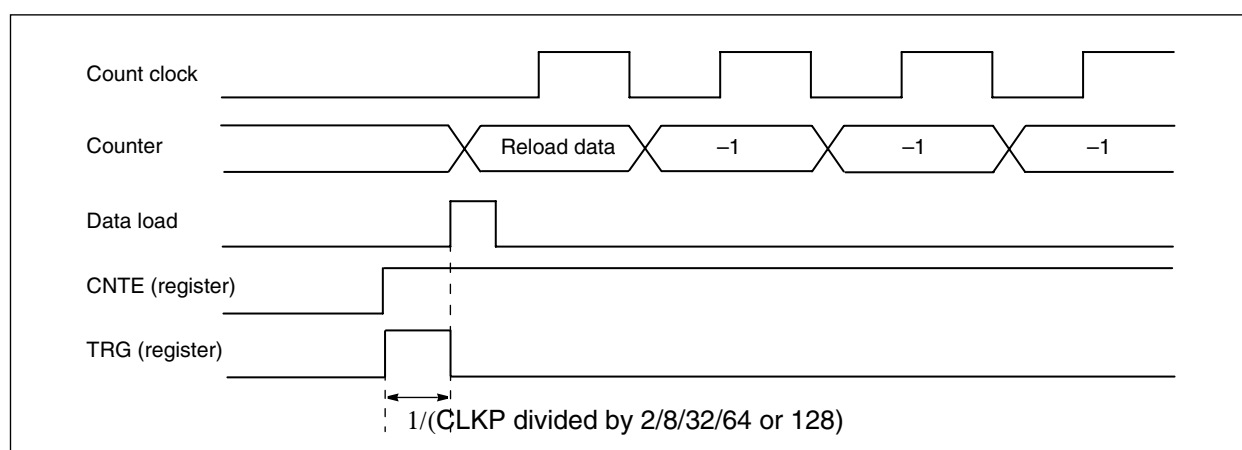
Using the TRG bit as a trigger input is always available when the counter is enabled (CNTE = "1"), regardless of the operation mode.

Figure 3-1 shows counter activation and counter operation.

One clock cycle (CLKP divided by 2/8/32/64 or 128) is required from the counter start trigger being input until the reload register data is loaded into counter.

● Counter activation and operation timing

Figure 40.3-1 Counter Activation and Operation Timing



40.3.2 Underflow Operation

An underflow occurs when the counter value changes from 0000H to FFFFH. Therefore, an underflow occurs after "reload register setting + 1" counts.

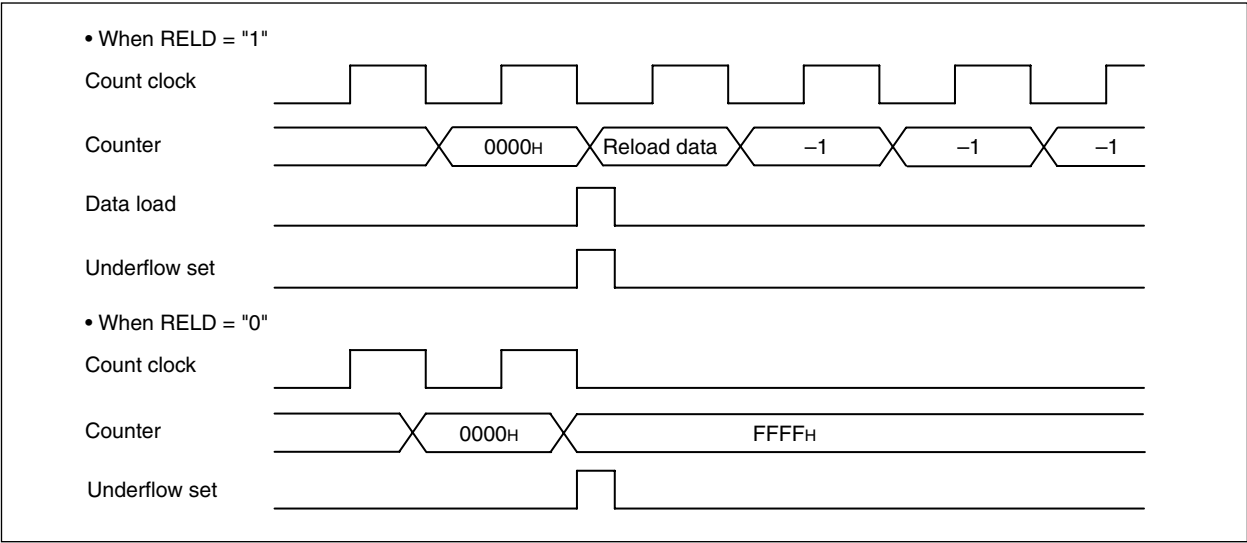
If the RELD bit in the control register is "1" when the underflow occurs, the contents of the reload register are loaded into the counter and counting continues. When RELD is "0", counting stops with the counter at FFFFH.

The UF bit in the control register is set when the underflow occurs. If the INTE bit is "1" at this time, an interrupt request is generated.

Figure 3-2 shows the operation when an underflow occurs.

● Underflow operation timing

Figure 40.3-2 Underflow Operation Timing



40.3.3 Counter Operation States

The counter state is determined by the CNTE bit in the control register and the internal WAIT signal. The available states are

CNTE = "0" and WAIT = "1" (STOP state: operation halted),

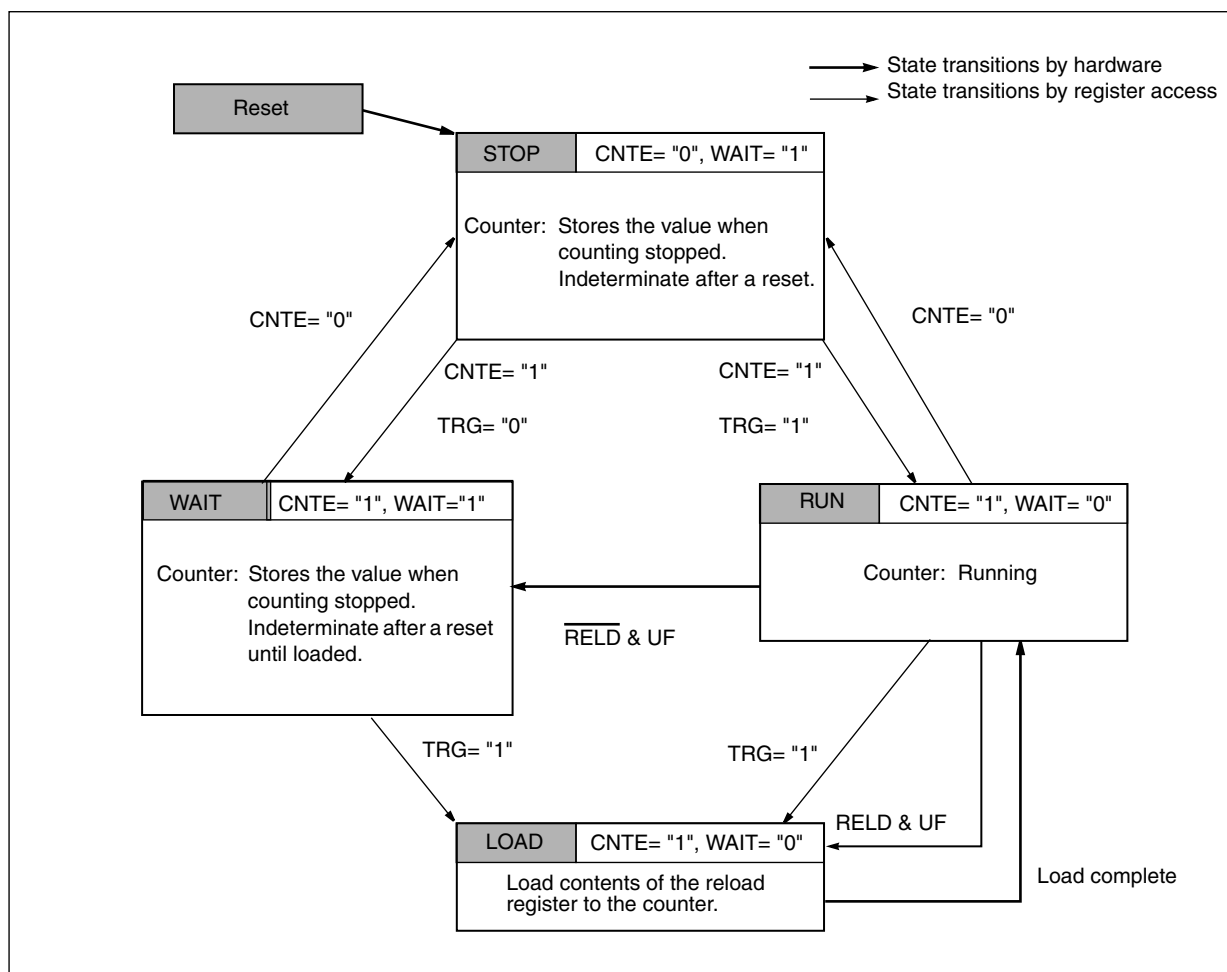
CNTE = "1" and WAIT = "1" (WAIT state: waiting for a trigger), and

CNTE = "1" and WAIT = "0" (RUN state: operating).

Figure 40.3-3 shows the transitions between each state.

● Counter state transitions

Figure 40.3-3 Counter State Transitions



40.4. PFM Operation and Setting

This section describes the following operations of the 16-bit pulse frequency modulator (combining the functionality of both reload counters).

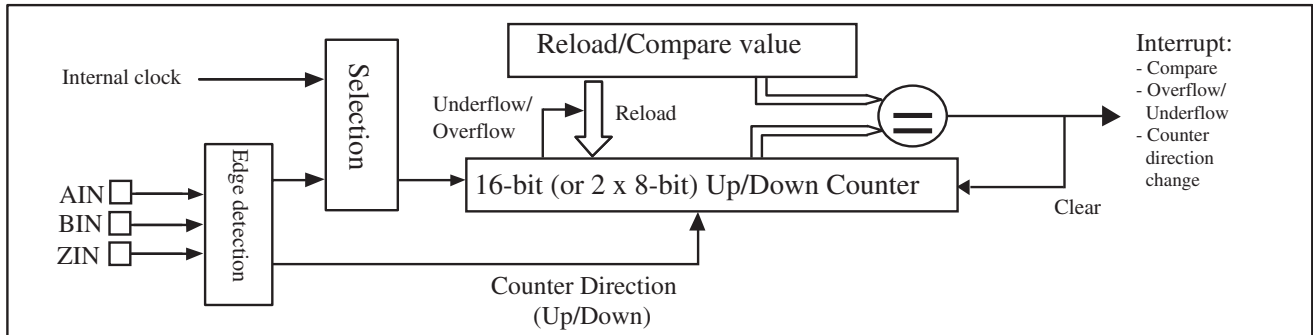
1. The underflow output of reload counter channel 0 is connected internally to the trigger input reload counter channel 1. The underflow output of reload counter channel 1 is connected internally to the trigger input of reload counter channel 0.
2. Both counters must be set up with $RELD = "0"$. Counter 0 should then be started by software trigger $TRG = "1"$. By starting counter 0 a high level is generated at the output. At the underflow condition of counter 0, counter 1 is automatically reloaded and started by the internal trigger (falling edge, set $MOD1 = "1"$ for both counters) and a low level is generated at the output. At the underflow condition of counter 1, counter 0 is automatically reloaded and started by the internal trigger and a high level is generated at the output.
3. Interrupts can be set up on underflow condition of counter 0, counter 1 or both. The interrupts of counter 0 and counter 1 are combined together (logical OR).
4. The default output is low level if both $CNTE = "0"$, and both $INV = "0"$.

Chapter 41 Up/Down Counter (UDC)

41.1. Overview

Triggered by an input signal, 16-bit Up/Down Counter counts up or down within the range of 0 to 65535. Specifically, Up/Down Counter running in the phase difference count mode is suitable for counting the encoder pulse of motors and other equipment. When encoder's output signals of phase A, phase B and phase Z are applied, the counter can achieve precise counting of rotation angles or number of revolutions.

Figure 41.1-1 Block diagram of an Up/Down Counter



41.2. Feature

- Format: 16 bit length or 8 bit x 2
- Quantity: 2 for 16 bit (Inputs: AIN0/BIN0/ZIN0, AIN2/BIN2/ZIN2)
4 for 8 bit (Inputs: AIN0/BIN0/ZIN0, AIN1/BIN1/ZIN1, AIN2/BIN2/ZIN2, AIN3/BIN3/ZIN3)

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Count mode: Four types
 - Timer mode
Count down the internal clock.
 - Up/down count mode
Counting up is triggered by an AIN pin signal.
Counting down is triggered by a BIN pin signal.
 - Phase difference count mode (Multiply by 2)
Counting is triggered by the rising and falling edges of a BIN pin signal. Up/Down Counter counts up or down, depending on the AIN pin signal level.
 - Phase difference count mode (Multiply by 4)
Counting is triggered by the rising and falling edges of AIN and BIN pin signals.
If an edge on AIN pin occurs, the Counter counts up or down, depending on the BIN pin signal level.
If an edge on BIN pin occurs, the Counter counts up or down, depending on the AIN pin signal level.
- Count Source
Internal clock (Timer mode): Peripheral clock (CLKP) divided by 2 or 8
External trigger (Up/down count mode): Edge detection (Rising/falling/both edges/no detection)
- Counting range: Any value between 0 and 65535 can be set.
- Interrupt: Select from the following four types:
 - (1) Compare-match interrupt
 - (2) Underflow interrupt

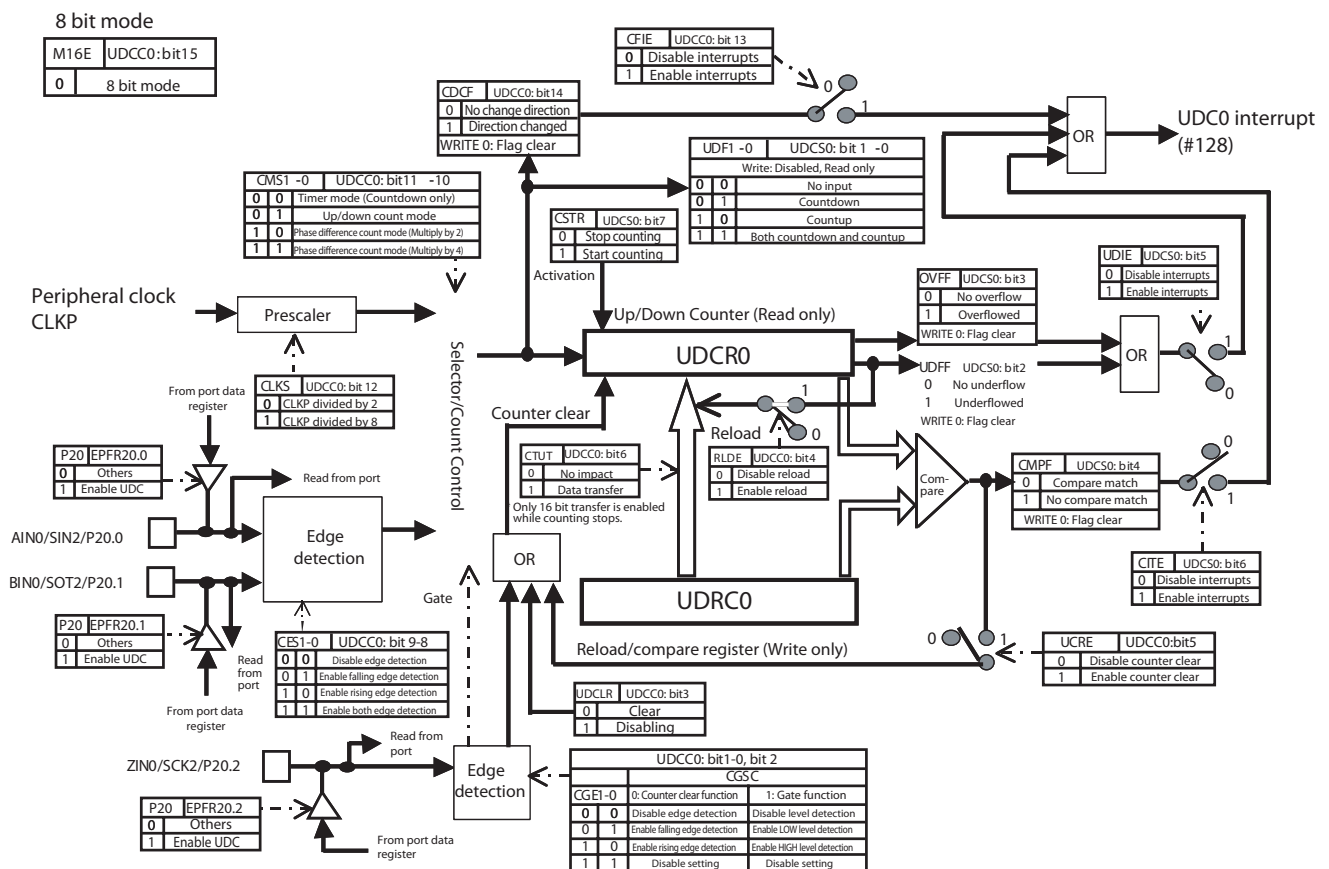
- Whether counting is performed or not can be controlled based on the pin input level.

The ZIN pin has two functions: Counter clear and gate.

The count direction flag allows identification of the previous count direction.

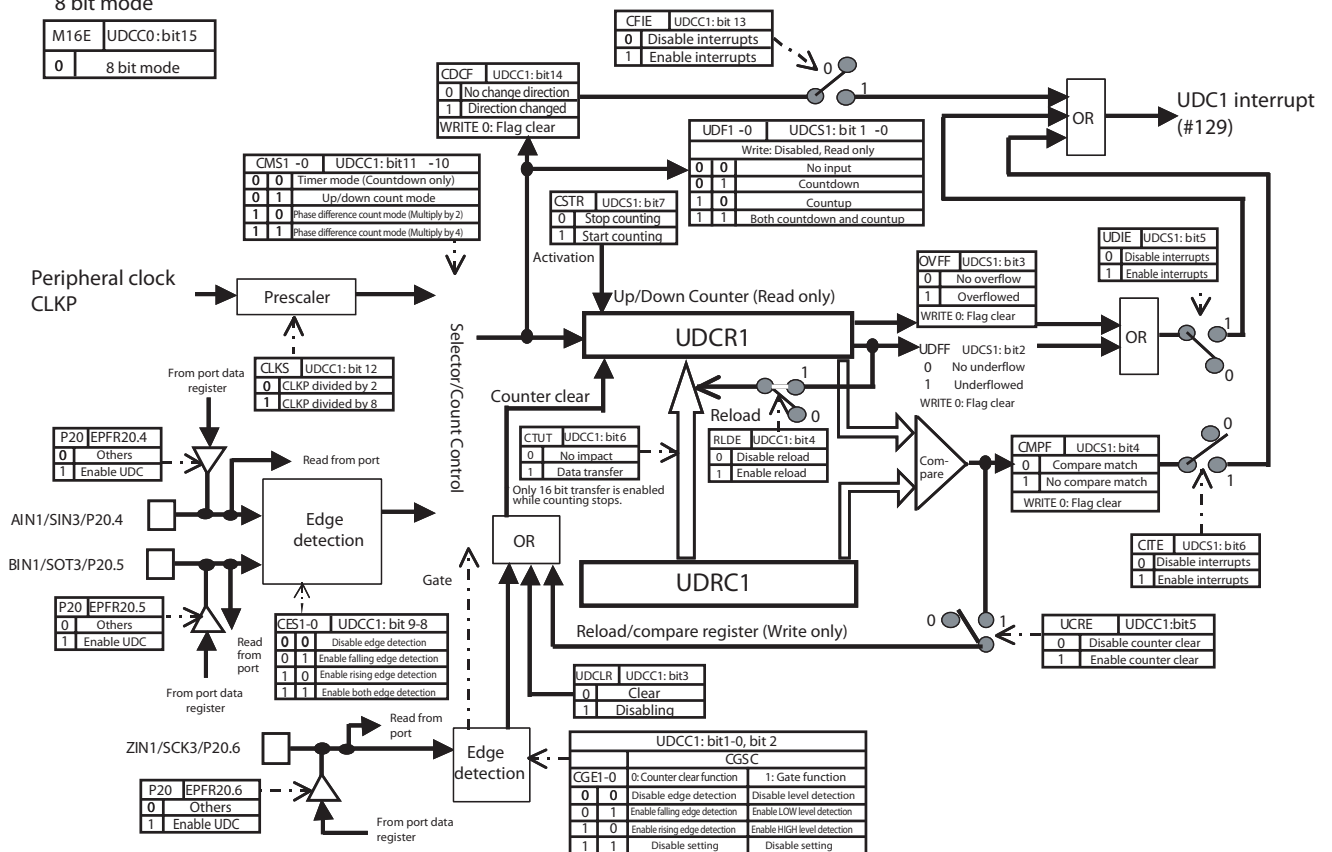
Figure 41.3-1 Configuration Diagram of the Up/Down Counter 0 (8 bit mode)

Up/Down Counter 0 (8 Bit Mode)



8 bit mode

M16E	UDCC0:bit15
0	8 bit mode



16 bit mode

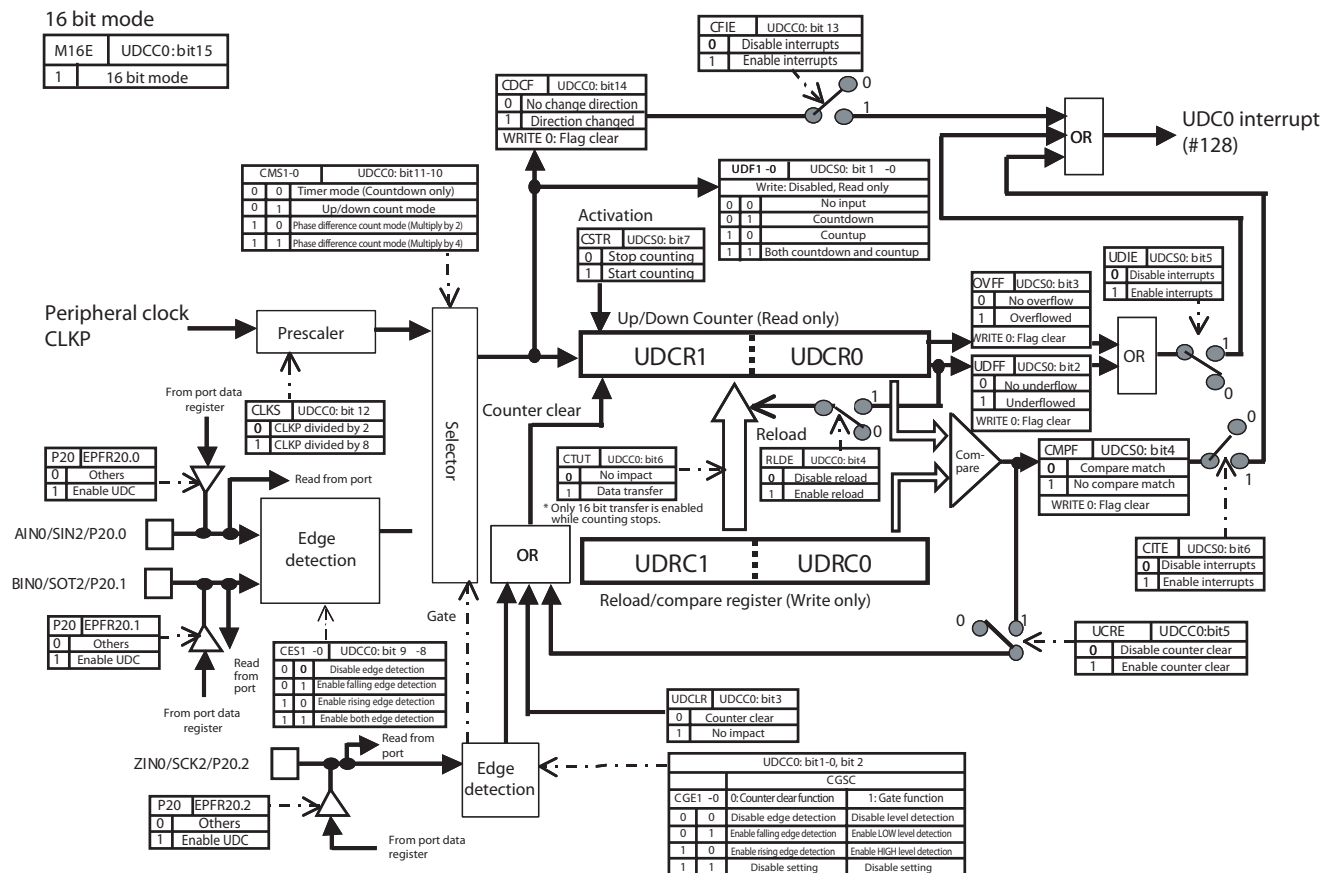


Figure 41.3-4 Up/Down Counter 0 Register List (8-bit mode)

Up/Down Counter 0 (8 bit mode)																		
Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000304H		M16E	CDLF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	UDCC0 (Counter control 0)
000301H	Bit	7	6	5	4	3	2	1	0									
		D7	D6	D5	D4	D3	D2	D1	D0	UDCR0	(Reload/Compare 0)							
000303H		D7	D6	D5	D4	D3	D2	D1	D0	UDCR0	(Up/Down Counter 0)							
000307H		CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	UDCS0	(Status 0)							
000D94H		-	SCK3	SOT3	SIN3	-	SCK2	SOT2	SIN2	PFR20	(Port function 20)							
000DD4H		-	ZIN1	BIN1	AIN1	-	ZIN0	BIN0	AIN0	EPFR20	(Extra port function 20)							
000478H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0	ICR56	(Interrupt level for Up/Down Counter)							
0FFDFCH	32Bits																(Interrupt Vector #128)	

^a For ICR registers and interrupt vectors, refer to the "Interrupt Control" section.

Note: For ICR registers and interrupt vectors, refer to “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

Figure 41.3-5 Up/Down Counter 1 Register List (8-bit mode)

Up/Down Counter 1 (8 bit mode)

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000304H		M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	UDCC0 (Counter control 0)
000308H		reserved	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	UDCC1 (Counter control 1)
000300H	Bit	7	6	5	4	3	2	1	0									
		D7	D6	D5	D4	D3	D2	D1	D0									UDRC1 (Reload/Compare 1)
000302H		D7	D6	D5	D4	D3	D2	D1	D0									UDCR1 (Up/Down Counter 1)
00030BH		CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0									UDCS1 (Status 1)
000D94H		-	SCK3	SOT3	SIN3	-	SCK2	SOT2	SIN2									PFR20 (Port function 20)
000DD4H		-	ZIN1	BIN1	AIN1	-	ZIN0	BIN0	AIN0									EPFR20 (Extra port function 20)
000478H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0									ICR56 (Interrupt level for Up/Down Counter)
0FFDF8H		32Bits																(Interrupt Vector #129)

* For ICR registers and interrupt vectors, refer to the "Interrupt Control" section.

Note: For ICR registers and interrupt vectors, refer to “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

Figure 41.3-6 Up/Down Counter 0/1 Register List (16-bit mode)

Up/Down Counter 0 (16 bit mode)

Address	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
000304H		M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	UDCC0 (Counter control 0)
000300H		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	UDRC (UDRC1,UDRC0) (Reload/Compare)
000302H		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	UDCR (UDCR1,UDCR0) (Up/Down Counter)
000307H	Bit	7	6	5	4	3	2	1	0									
		CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0									UDCS0 (Status 0)
000D94H		-	SCK3	SOT3	SIN3	-	SCK2	SOT2	SIN2									PFR20 (Port function 20)
000DD4H		-	ZIN1	BIN1	AIN1	-	ZIN0	BIN0	AIN0									EPFR20 (Extra port function 20)
000478H		---	---	---	ICR4	ICR3	ICR2	ICR1	ICR0									ICR56 (Interrupt level for Up/Down Counter)
0FFDFCH		32Bits																(Interrupt Vector #128)

* For ICR registers and interrupt vectors, refer to the "Interrupt Control" section.

Note: For ICR registers and interrupt vectors, refer to “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

41.4. Register

41.4.1 UDCC: Counter Control Register

This register is used to control behaviors of Up/Down Counter.

- **UDCC0 (Up/Down Counter 0): Address 0304_H (Access: Byte, Half-word)**
- **UDCC1 (Up/Down Counter 1): Address 0308_H (Access: Byte, Half-word)**
- **UDCC2 (Up/Down Counter 2): Address 0314_H (Access: Byte, Half-word)**
- **UDCC3 (Up/Down Counter 3): Address 0318_H (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	bit
M16E/ Reserved	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	UDCCH
0	0	0	0	0	0	0	0	Initial value
R/W *	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	bit
Reserved	CTUT	UCRE	RLDE	UDCLR	CGSC	CGE1	CGE0	UDCCL
0	0	0	0	1	0	0	0	Initial value
R/W0	R/W	R/W	R/W	R1,W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit15: Enable 16 bit mode (Up/Down Counter 0 and 2 only)

M16E	Enable 16 bit mode
0	8 bit × 2 channel operation mode (8 bit mode)
1	16 bit × 1 channel operation mode (16 bit mode)

* Reserved bit (Up/Down Counter 1 and 3). Be sure to write 0. The read value is the value written.

- bit14: Count direction change flag (Interrupt flag)

CDCF	Direction change detection	
	When read:	When written:
0	No direction change occurred.	Clear the flag.
1	A direction change occurred once or more.	No effect.

- When the count direction has been changed during count operation, the count direction change flag (CDCF) is set to “1”.
- Since the count direction is set to countdown immediately after a reset, the count direction change flag (CDCF) is set to “1” on counting up following the reset.
- To enable interrupt requests, the interrupt request enable bit must be set (CFIE=“1”).
- bit13: Enable count direction change interrupt request

CFIE	Direction change interrupt request
0	Disable direction change interrupt requests.
1	Enable direction change interrupt requests.

MB91460 Series

When the interrupt request enable bit is set to “1”, the interrupt flag (CDCF) is enabled.

- bit12: Select internal prescaler

CLKS	Internal clock frequency
0	CLKP/2
1	CLKP/8

This setting is enabled only in the timer mode, in which only countdown is performed.

- bit11,10: Select count mode

CMS1	CMS0	Count mode
0	0	Timer mode (Countdown)
0	1	Up/down count mode
1	0	Phase difference count mode (Multiply by 2)
1	1	Phase difference count mode (Multiply by 4)

- bit9,8: Select count clock edge

CES1	CES0	Edge selection
0	0	Disable edge detection.
0	1	Detect a falling edge.
1	0	Detect a rising edge.
1	1	Detect both rising and falling edges.

This bit is used in the up/down count mode (CMS1,CMS0= “01”) to select the edge, to be detected, of an AIN and BIN pin signal. This setting is disabled in modes other than the up/down count.

- bit7: Reserved.
Be sure to write “0”. The read value is the value written.
- bit6: Counter write

CTUT	Data transfer
0	No impact on operation
1	Transfer data from the RCR register to UDCR.

During count operation (CSR.CSTR=“1”), the counter write bit must not be set to “1”.

- bit5: Enable compare-match clear

UCRE	Compare-match counter clear
0	Disable counter clear due to compare-match.
1	Enable counter clear due to compare-match.

This setting does not affect clear operations other than compare-match, such as ZIN pin clear.

- bit4: Enable reload

RLDE	Reload function
0	Disable reload function.
1	Enable reload function.

If the reload enable bit is set to “1”, the reload/compare value (RCR) is transferred to Up/Down Counter (UDCR) when Up/Down Counter is underflowed.

- bit3: Clear UDCR

UDCLR	Counter clear
0	Set (Clear) Up/Down Counter (UDCR) to “0000H”.
1	No impact on operation

- bit2: Select counter clear/gate

CGSC	ZIN pin function
0	Counter clear function
1	Gate function

- bit1,0: Select counter clear/gate edge

CGE1	CGE0	Edge detection/level selection	
		When the counter clear function is selected (CGSC=“0”)	When the gate function is selected (CGSC=“1”)
0	0	Disable edge detection.	Disable level detection. (Disable count.)
0	1	Detect a falling edge.	Detect a “L” level.
1	0	Detect a rising edge.	Detect a “H” level.
1	1	Disable setting.	Disable setting.

41.4.2 UDCS: Count Status Register

This register is used to control Up/Down Counter and to indicate the status of the counter.

- **UDCS0 (Up/Down Counter 0): Address 0307_H (Access: Byte, Half-Word)**
- **UDCS1 (Up/Down Counter 1): Address 030B_H (Access: Byte, Half-Word)**
- **UDCS2 (Up/Down Counter 2): Address 0317_H (Access: Byte, Half-Word)**
- **UDCS3 (Up/Down Counter 3): Address 031B_H (Access: Byte, Half-Word)**

7	6	5	4	3	2	1	0	bit
CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W0	R/W0	R/W0	R/WX	R/WX	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit7: Enable count operation

CSTR	Count operation
0	Disable count operation.
1	Enable count operation. (Activate counter.)

- bit6: Enable compare-match interrupt requests

CITE	Compare-match interrupt request
0	Disable compare-match interrupt requests.
1	Enable compare-match interrupt requests.

Setting the interrupt request enable bit to “1” enables the interrupt flag (CMPF).

- bit5: Enable overflow/underflow interrupt request

UDIE	Overflow/underflow interrupt request
0	Disable overflow/underflow interrupt requests.
1	Enable overflow/underflow interrupt requests.

Setting the interrupt request enable bit to “1” enables the interrupt flag (OVFF or UDFF).

- bit4: Compare-match detection flag

CMPF	Compare-match detection	
	When read:	When written:
0	Comparison results do not match.	Clear the flag.
1	Comparison results match.	Without effect.

To enable interrupt requests, the interrupt request enable bit must be set (CITE= “1”).

- bit3: Overflow detection flag

OVFF	Overflow detection	
	When read:	When written:
0	No overflow	Clear the flag.
1	An overflow has occurred.	Without effect.

To enable interrupt requests, the interrupt request enable bit must be set (UDIE= “1”).

- bit2: Underflow detection flag

UDFF	Underflow detection	
	When read:	When written:
0	No Underflow	Clear the flag.
1	An underflow has occurred.	Without effect.

To enable interrupt requests, the interrupt request enable bit must be set (UDIE= “1”).

- bit1,0: Up/down flag

UDF1	UDF0	Previous count operation
0	0	No input
0	1	Count down
1	0	Count up
1	1	Both of count up and count down

MB91460 Series

41.4.3 UDCR: Up/Down Counter Register

This register is used to read the count value of Up/Down Counter.

- **UDCR10 (Up/Down Counter 0/1): Address 0302_H (Access: Byte, Half-Word)**
- **UDCR32 (Up/Down Counter 2/3): Address 0312_H (Access: Byte, Half-Word)**

Depending on the setting of the 16-bit mode enable bit (CCR.M16E), this register behaves differently.

■ 16 Bit Mode (M16E= “1”)

In the 16 bit mode, this register functions as 16-bit up/down counter register.

- UDCR10 (Up/Down Counter): Address 0302_H (Access: Half-word)
- UDCR32 (Up/Down Counter): Address 0312_H (Access: Half-word)

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D09	D08	
0	0	0	0	0	0	0	0	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	bit
D07	D06	D05	D04	D03	D02	D01	D00	
0	0	0	0	0	0	0	0	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

For the 16 bit mode, be sure to read by half-word access.

■ 8 Bit Mode (M16E=“0”)

In the 8 bit mode, this register functions as a 8-bit up/down counter register 0 and a 8-bit up/down counter register 1.

- UDCR1 (Up/Down Counter 1): Address 00302_H (Access: Byte, Half-word)
- UDCR0 (Up/Down Counter 0): Address 00303_H (Access: Byte, Half-word)
- UDCR3 (Up/Down Counter 3): Address 00312_H (Access: Byte, Half-word)
- UDCR2 (Up/Down Counter 2): Address 00313_H (Access: Byte, Half-word)

7	6	5	4	3	2	1	0	bit
D07	D06	D05	D04	D03	D02	D01	D00	
0	0	0	0	0	0	0	0	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

41.4.4 UDRC: Up/Down Reload/Compare Register

This register is used to reload a value to Up/Down Counter and to compare the register value with the Counter value.

This register is also used to write to Up/Down Counter.

- **UDRC10 (Up/Down Counter 0/1): Address 0300_H (Access: Byte, Half-Word)**
- **UDRC32 (Up/Down Counter 2/3): Address 0310_H (Access: Byte, Half-Word)**

Depending on the setting of the 16 bit mode enable bit (CCR.M16E), this register behaves differently.

■ 16 Bit Mode (M16E="1")

In the 16 bit mode, this register functions as 16-bit reload/compare register.

- UDRC10 (Reload compare): Address 00300_H (Access: Byte, Half-word)
- UDRC32 (Reload compare): Address 00310_H (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D09	D08	
0	0	0	0	0	0	0	0	Initial value
RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	Attribute

7	6	5	4	3	2	1	0	bit
D07	D06	D05	D04	D03	D02	D01	D00	
0	0	0	0	0	0	0	0	Initial value
RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- The reload and compare values are same.

When Up/Down Counter counts up, the value in RCR is used as a compare value.

When Up/Down Counter counts down, underflow is generated and the value in RCR is used as a reload value for reloading.

(Up/Down Counter counts between 0000_H and the reload/compare value.)

- In the 16 bit mode, be sure to write by half-word access.

■ 8 Bit Mode (M16E="0")

In the 8 bit mode, this register functions as 8-bit reload/compare register 0 and 8-bit reload/compare register 1.

- UDRC1 (reload compare 1): Address 00300_H (Access: Byte, Half-word)
- UDRC0 (reload compare 0): Address 00301_H (Access: Byte, Half-word)
- UDRC3 (reload compare 3): Address 00310_H (Access: Byte, Half-word)
- UDRC2 (reload compare 2): Address 00311_H (Access: Byte, Half-word)

7	6	5	4	3	2	1	0	bit
D07	D06	D05	D04	D03	D02	D01	D00	
0	0	0	0	0	0	0	0	Initial value
RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	RX, W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).)

- The reload and compare values are same.

When Up/Down Counter counts up, the value in RCR is used as a compare value.

When Up/Down Counter counts down, underflow is generated and the value in RCR is used as a reload value for reloading.

(Up/Down Counter counts between 00_H and the reload/compare value.)

- Perform the following procedure to write to Up/Down Counter.

(1) Stop counting.

(2) Write a value to the reload/compare register.

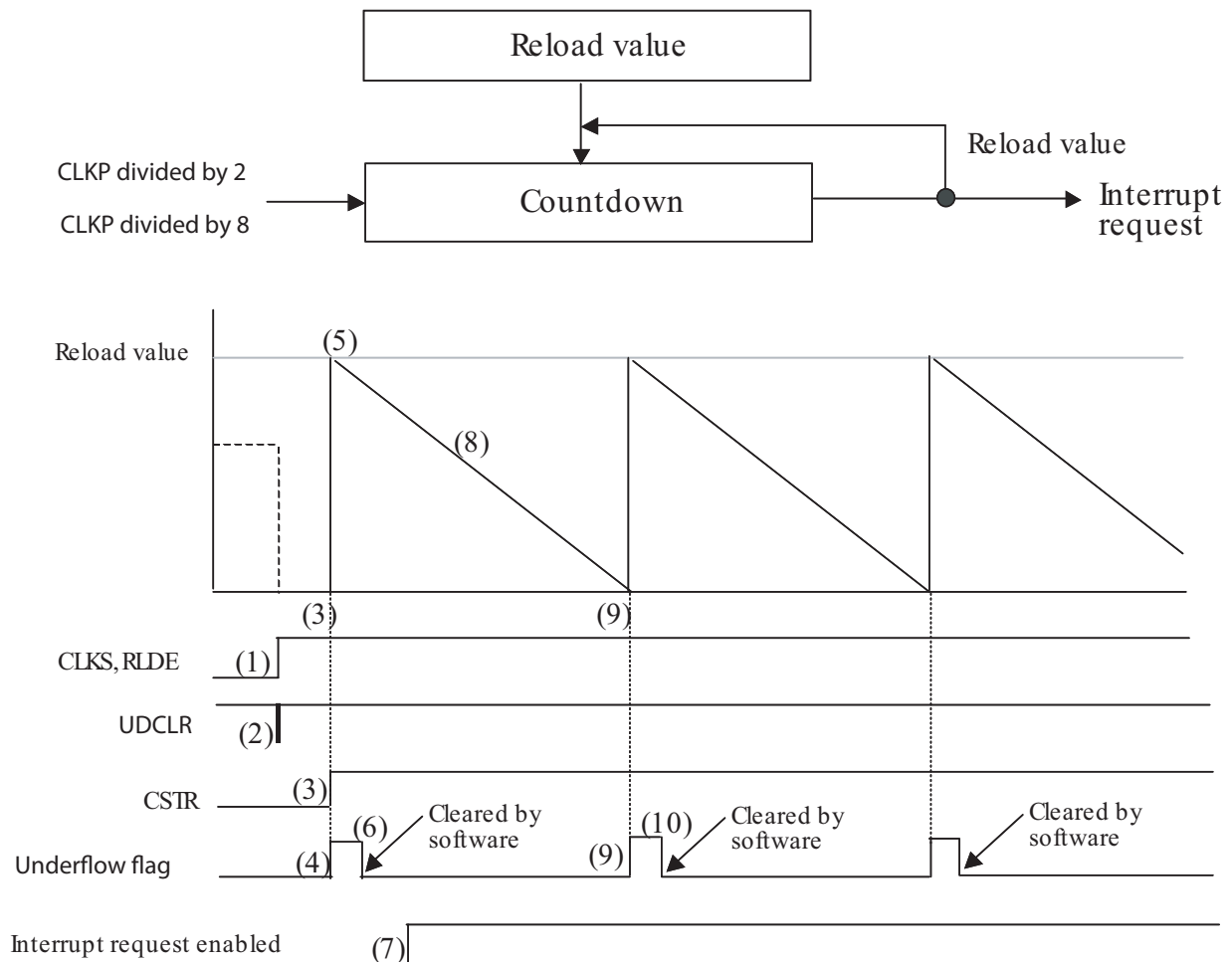
(3) Write “1” to the counter write bit (CCR.CTUT).

41.5. Operation

This section describes each operation mode for Up/Down Counter.

41.5.1 Timer Mode CMS[1:0]="00"

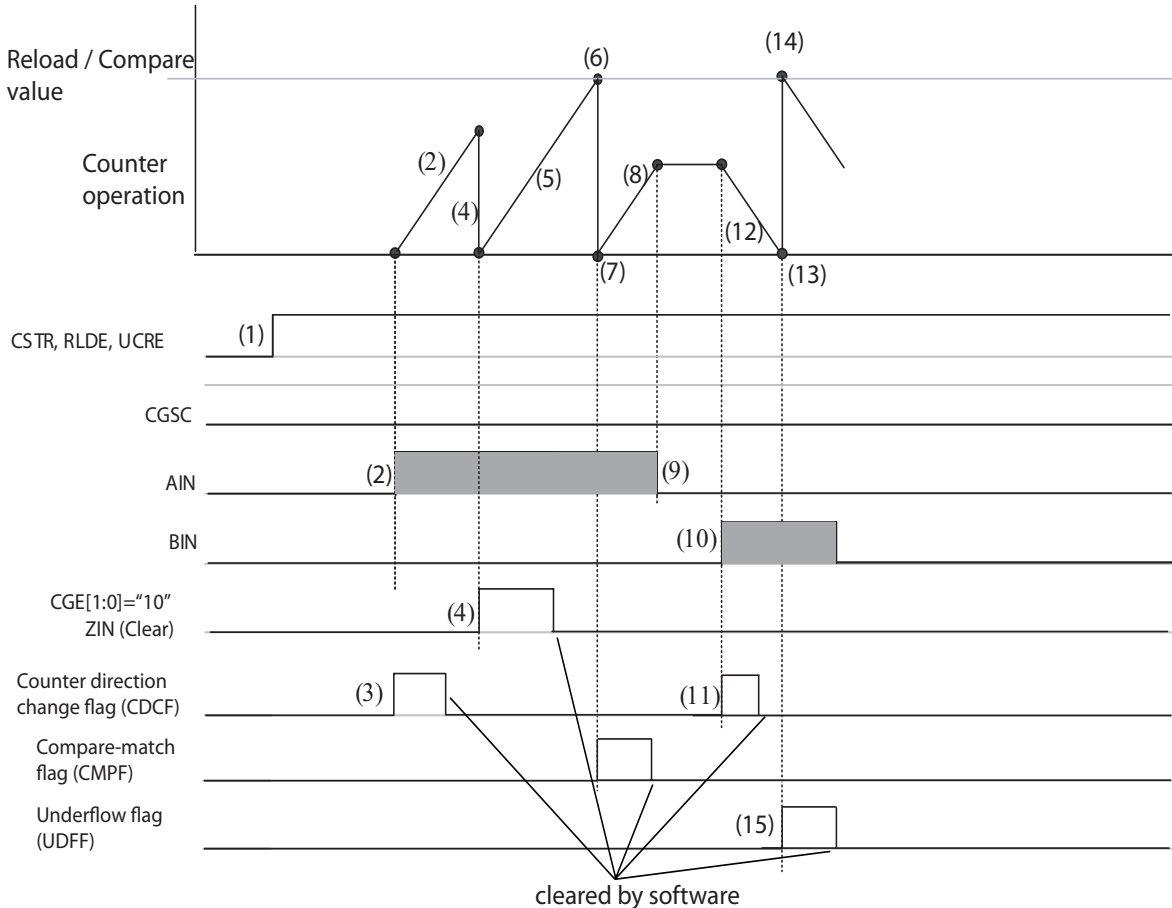
Figure 41.5-1 Up/Down Counter in Timer Mode CMS[1:0]="00"



- (1) An appropriate bit (Reload enable RLDE) is set.
- (2) Up/Down Counter is cleared ("0" is written to UDCLR).
- (3) The software activates Up/Down Counter.
- (4) An underflow occurs.
- (5) The reload value is reloaded to Up/Down Counter.
- (6) The software clears the underflow flag.
- (7) The software enables interrupts.
- (8) Up/Down Counter counts down.
- (9) An underflow occurs. (An interrupt request has been made.)
- (10) The software clears the underflow flag.
- (11) Repeat (8) to (10).

MB91460 Series**41.5.2 Up/Down Count Mode CMS[1:0]="01"****Figure 41.5-2 Up/Down Counter in Up/Down Count Mode CMS[1:0]="01"**

ZIN=Clear control



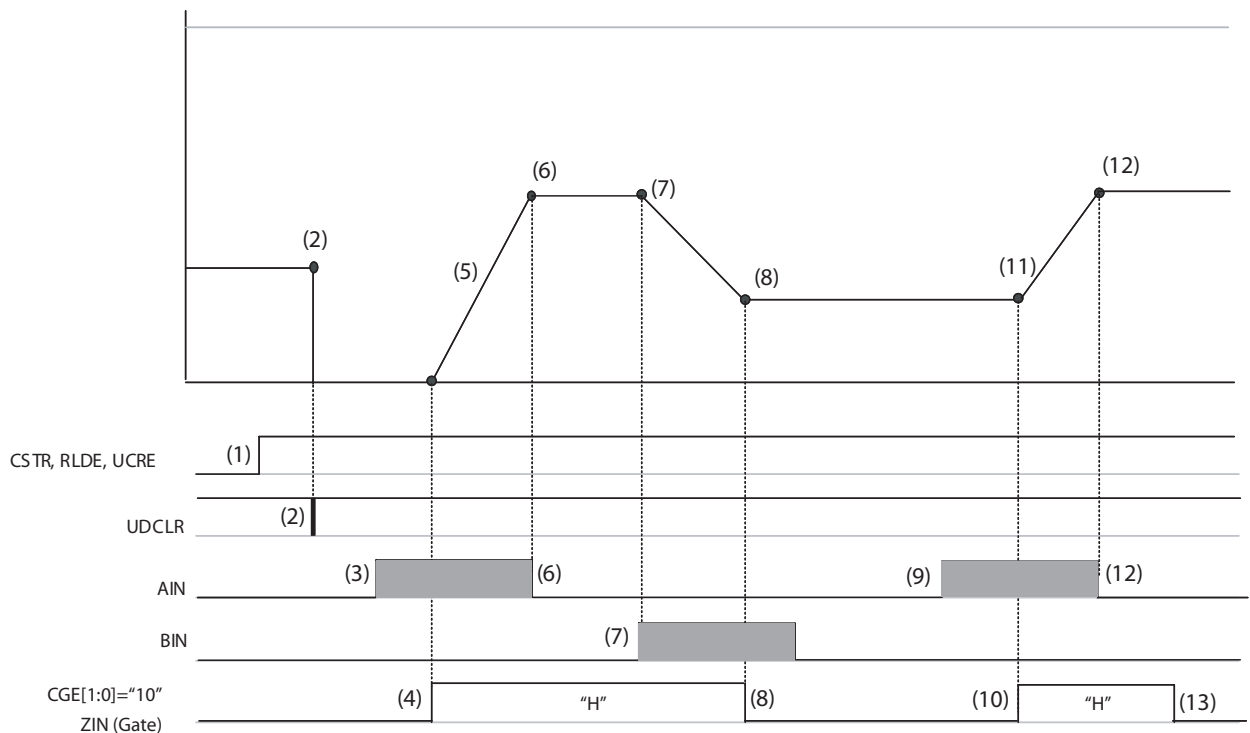
Up/Down Counter clear control using the ZIN pin

- (1) Appropriate bits (Counting enable CSTR, Reload enable RLDE, Clear enable UCRE) are set.
- (2) When pulse input to the AIN pin is detected, Up/Down Counter counts up.
- (3) The count direction change flag is set to "1".
- (4) When an edge is applied to the ZIN pin, Up/Down Counter is cleared.
- (5) Continuous pulse input to the AIN pin causes Up/Down Counter to count up.
- (6) The Up/Down Counter's count value matches with the compare value (compare-match) and the compare-match flag is set to "1".
- (7) Compare-match clears Up/Down Counter.
- (8) Continuous pulse input to the AIN pin causes Up/Down Counter to count up.
- (9) When pulse input to the AIN pin stops, Up/Down Counter stops counting.
- (10) When pulse input to the BIN pin is detected, Up/Down Counter counts down.
- (11) The count direction change flag is set to "1".
- (12) Continuous pulse input to the BIN pin causes Up/Down Counter to count down.
- (13) Up/Down Counter is underflowed and the underflow flag is set to "1".
- (14) The underflow causes the reload value to be reloaded to Up/Down Counter.
- (15) Next time when Up/Down Counter counts down, the compare-match flag is set to "1".

41.5.3 Up/Down Count Mode CMS[1:0]="01"

Figure 41.5-3 Up/Down Counter in Up/Down Count Mode CMS[1:0]="01"

ZIN=Gate control



Countgate at the ZIN pin

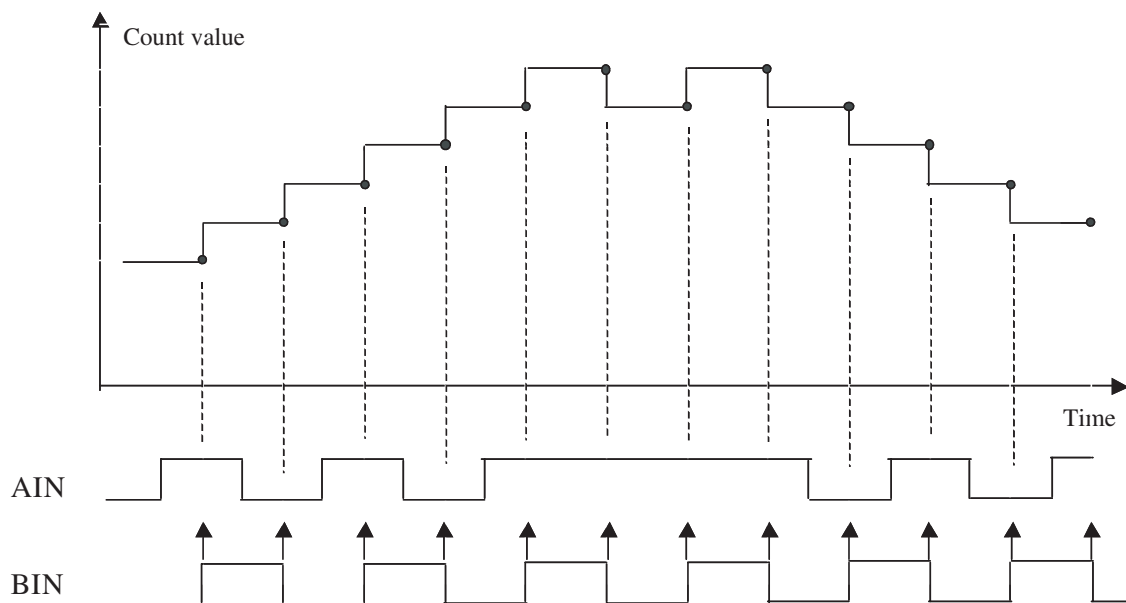
- (1) Appropriate bits (Counting enable CSTR, Reload enable RLDE and Clear enable UCRE) are set.
- (2) Up/Down Counter is cleared. ("0" is written to UDCLR).
- (3) Neither pulse input to the AIN pin nor counting at the ZIN pin being enabled, Up/Down Counter neither counts up nor down.
- (4) Counting is enabled at the ZIN pin.
- (5) Up/Down Counter counts up.
- (6) When pulse input to the AIN pin stops, Up/Down Counter stops counting.
- (7) When a pulse input to the BIN pin is detected, Up/Down Counter counts down.
- (8) When counting is disabled at the ZIN pin, Up/Down Counter stops counting.
- (9) Neither pulse input to the AIN pin nor counting at the ZIN pin being enabled, Up/Down Counter neither counts up nor down.
- (10) Counting is enabled at the ZIN pin.
- (11) Up/Down Counter counts up.
- (12) When pulse input to the AIN pin stops, Up/Down Counter stops counting.
- (13) Counting is disabled at the ZIN pin.

41.5.4 Phase Difference Count Mode (Multiply by 2) CMS[1:0]="10"

Frequency multiplied by 2 in phase difference count mode:

On the rising and falling edges at the BIN count pin, Up/Down Counter counts up or down, depending on the voltage level at the AIN pin.

Figure 41.5-4 Up/Down Counter in Phase Difference Count Mode (Multiply by 2) CMS[1:0]="10"



- Count up Conditions:
 - When the voltage level at the AIN pin detected on the rising edge at the BIN pin is "H"
 - When the voltage level at the AIN pin detected on the falling edge at the BIN pin is "L"
- Count down Conditions:
 - When the voltage level at the AIN pin detected on the rising edge at the BIN pin is "L"
 - When the voltage level at the AIN pin detected on the falling edge at the BIN pin is "H"

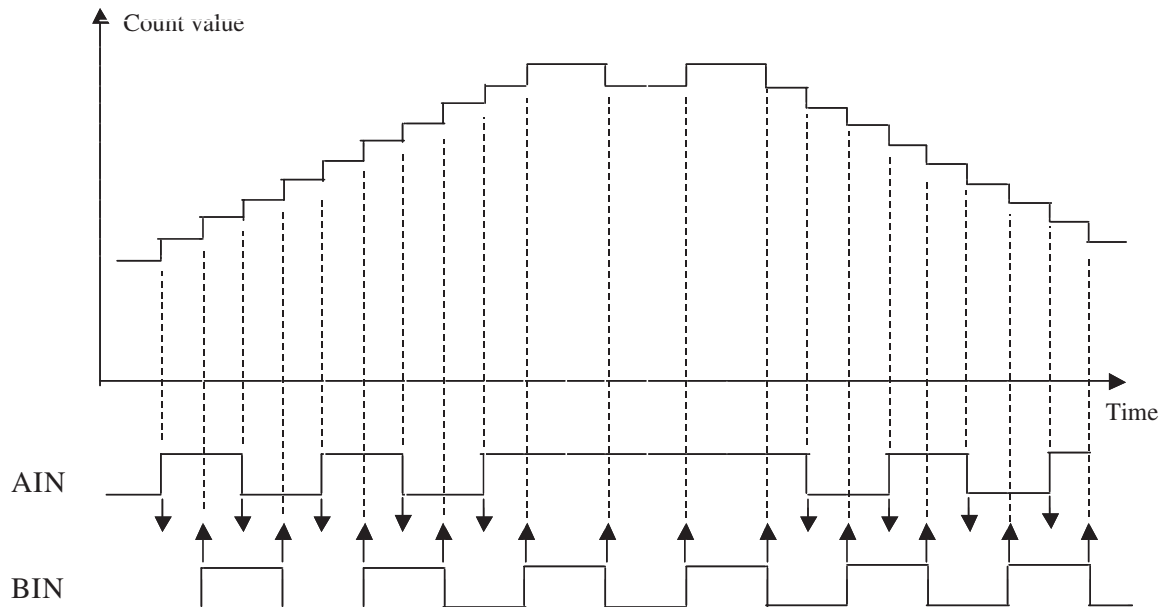
When this count mode is selected, selection of the edge to be detected using CES1 or CES0 is disabled.

41.5.5 Phase Difference Count Mode (Multiply by 4) CMS[1:0]="11"

Frequency multiplied by 4 in phase difference count mode:

On the rising and falling edges at the BIN pin, Up/Down Counter counts up or down, depending on the voltage level at the AIN pin, and on the rising and falling edges at the AIN pin, Up/Down Counter counts up or down, depending on the voltage level at the BIN pin.

Figure 41.5-5 Up/Down Counter in Phase Difference Count Mode (Multiply by 4) CMS[1:0]="11"



- Count up Conditions:

- When the voltage level at the AIN pin detected on the rising edge at the BIN pin is "H"
- When the voltage level at the AIN pin detected on the falling edge at the BIN pin is "L"
- When the voltage level at the BIN pin detected on the rising edge at the AIN pin is "L"
- When the voltage level at the BIN pin detected on the falling edge at the AIN pin is "H"

- Count down Conditions:

- When the voltage level at the AIN pin detected on the rising edge at the BIN pin is "L"
- When the voltage level at the AIN pin detected on the falling edge at the BIN pin is "H"
- When the voltage level at the BIN pin detected on the rising edge at the AIN pin is "H"
- When the voltage level at the BIN pin detected on the falling edge at the AIN pin is "L"

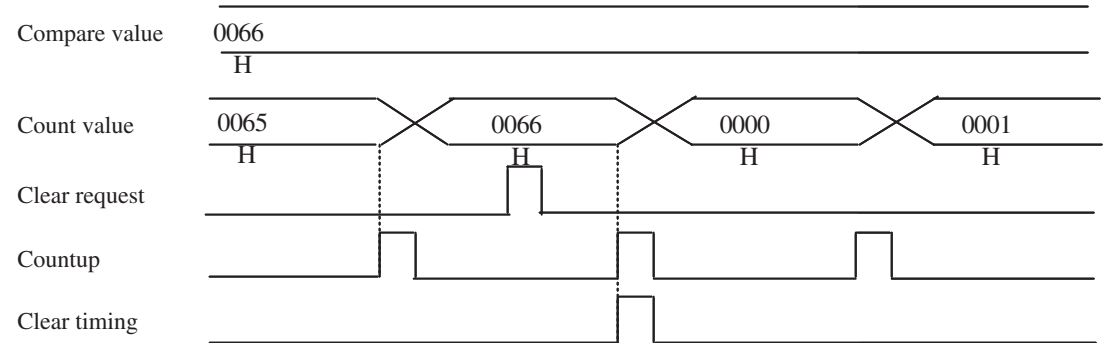
When Up/Down Counter is used to count encoder output, high precise counting of rotation angles and number of revolutions, as well as detecting of rotation directions, can be achieved by applying encoder output signals of phase A and phase B to the AIN and BIN, respectively.

Note that when this count mode is selected, selection of the edge to be detected using CES1 or CES0 is disabled.

41.5.6 Clear Timing

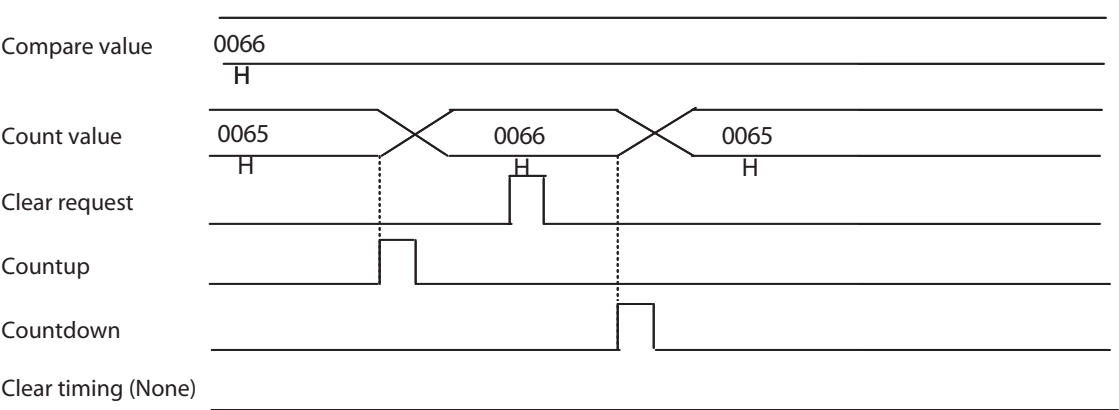
(1) When a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit UDCLR) is made, clear is performed next time when Up/Down Counter counts up.

Figure 41.5-6 Clear Timing of Up/Down Counter (1)



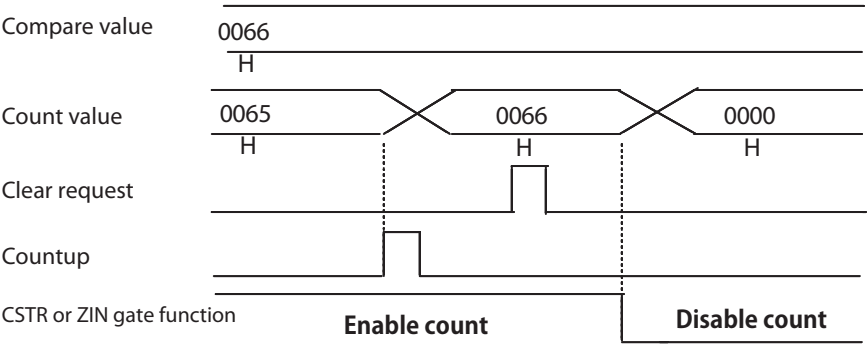
(2) Even if a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit DCC) is made, clear is not performed when UP/Down Counter counts neither up nor down.

Figure 41.5-7 Clear Timing of Up/Down Counter (2)



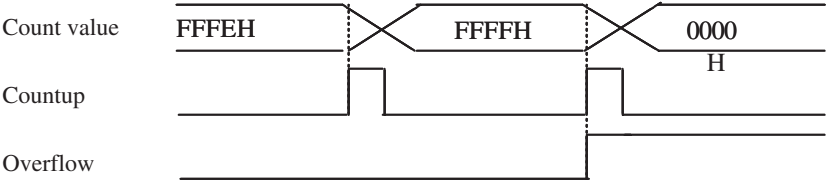
(3) If Up/Down Counter does not count up after a clear request (Compare-match, ZIN edge detection and writing “0” to the clear bit DCC) is made, the counter is cleared when counting is disabled (CSTR=“0”).

Figure 41.5-8 Clear Timing of Up/Down Counter (3)



(4) When Up/Down Counter exceeds the maximum count, the overflow flag is set to “1” and the counter value is returned to “0000”.

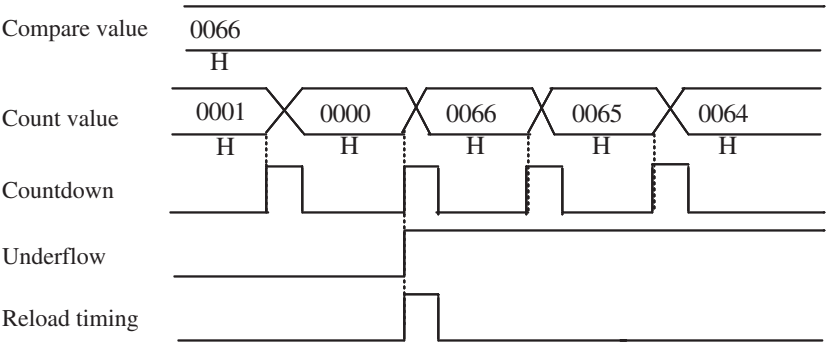
Figure 41.5-9 Clear Timing of Up/Down Counter (4)



41.5.7 Reload Timing

The next time when Up/Down Counter counts down below “0000”, an underflow occurs (an interrupt request is generated) and then reloading is performed.

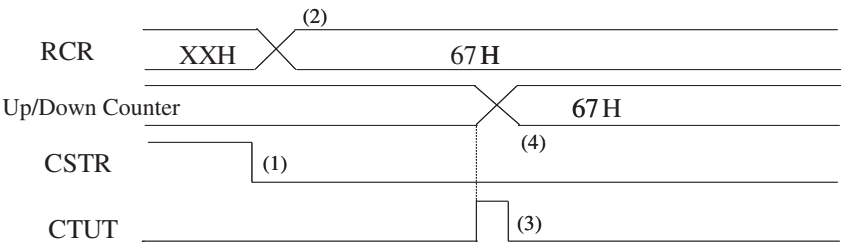
Figure 41.5-10 Reload Timing of Up/Down Counter



Note: If clear and reload operations occur at the same time, clear takes precedence.

41.5.8 Writing a Value to Counter

Figure 41.5-11 Writing value to Up/Down Counter



- (1) Counting of Up/Down Counter is disabled.
- (2) A value is written to PCR.
- (3) “1” is written to the count write bit CTUT.
- (4) A value is transferred from the reload/compare register RCR to Up/Down Counter.

MB91460 Series**41.6. Setting****Table 41.6-1 Required Settings to Run Up/Down Counter in Timer Mode**

Setting	Setting registers	Setting procedure*
Set the reload value.	Reload/compare register (UDRC)	See 41.7.16
(Optional) Set a value to Up/Down Counter or Clear the count value of Up/Down Counter.	Reload/compare register (UDRC)	See 41.7.5
	Count control register (UDCC)	See 41.7.8
Set a bit length.	Count control register (UDCC)	See 41.7.1
Set the count mode to timer mode.		See 41.7.2
Select a count source.		See 41.7.3
Enable reloading at the time of underflow.		See 41.7.7
Enable count control (clear/gate) using the ZIN pin.		See 41.7.9 and 41.7.10
Activate Up/Down Counter.	Count status register (UDCS)	See 41.7.11

*: For the setting procedure, refer to the section indicated by the number.

Table 41.6-2 Required Settings to Run Up/Down Counter in Up/Down Count Mode

Setting	Setting registers	Setting procedure*
Set the reload value/compare value.	Reload/compare register (UDRC)	See 41.7.16
(Optional) Set a value to Up/Down Counter Or Clear the count value of Up/Down Counter.	Reload/compare register (UDRC)	See 41.7.5
	Count control register (UDCC)	See 41.7.8
Set a bit length.	Count control register (UDCC)	See 41.7.1
Set the count mode to up/down count mode.		See 41.7.2
Select the edge, to be detected, of a signal (AIN or BIN), for which counting is performed.		See 41.7.4
Enable clearing of Up/Down Counter at the time of the counting following a compare-match.		See 41.7.6
Enable reloading at the time of underflow.		See 41.7.7
Enable count control (clear/gate) using the ZIN pin.		See 41.7.9 and 41.7.10
Activate Up/Down Counter.	Count status register (UDCS)	See 41.7.11

*: For the setting procedure, refer to the section indicated by the number.

Table 41.6-3 Required Settings to Run Up/Down Counter in Phase Difference Count Mode

Setting	Setting registers	Setting procedure*
Set the reload value/compare value.	Reload/compare register (UDRC).	See 41.7.16
(Optional) Set a value to Up/Down Counter or Clear the count value of Up/Down Counter.	Reload/compare register (UDRC)	See 41.7.5
	Count control register (UDCC)	See 41.7.8
Set a bit length.	Count control register (UDCC)	See 41.7.1
Set the count mode to phase difference count mode (Multiply by 2 or 4).		See 41.7.2
Enable clearing of Up/Down Counter at the time of the counting following a compare-match.		See 41.7.6
Enable reloading at the time of underflow.		See 41.7.7
Enable count control (clear/gate) using the ZIN pin.		See 41.7.9 and 41.7.10
Activate Up/Down Counter.	Count status register (UDCS)	See 41.7.11

*: For the setting procedure, refer to the section indicated by the number.

Table 41.6-4 Required Settings for Up/Down Counter Interrupt

Setting	Setting registers	Setting procedure*
Set Up/Down Counter interrupt vectors and Up/Down Counter interrupt levels.	Refer to “ Chapter 24 Interrupt Control (Page No.429) ”.	See 41.7.17
Set Up/Down Counter interrupts. Clear interrupts. Enable interrupt requests.	Count control register (UDCC) Count status register (UDCS)	See 41.7.19

*: For the setting procedure, refer to the section indicated by the number.

Table 41.6-5 Required Settings to Deactivate Up/Down Counter

Setting	Setting registers	Setting procedure*
Deactivate Up/Down Counter (Controlled through the ZIN pin)	Count control register (UDCC)	See 41.7.10
Deactivate Up/Down Counter.	Count status register (UDCS)	See 41.7.11

*: For the setting procedure, refer to the section indicated by the number.

MB91460 Series**41.7. Q&A****41.7.1 How to select a bit length (8 or 16) of Up/Down Counter**

Use the 16 bit mode enable bit (UDCC.M16E).

Up/Down Counter's bit length	16 bit mode enable bit (M16E)
To set the bit length to 8	Set the bit to "0".
To set the bit length to 16 bit	Set the bit to "1".

41.7.2 What types of count modes are available and how are they set?

There are four types of count modes:

Timer, Up/down count, Phase difference count (Multiply by 2 or 4)

Use the count mode selection bits (UDCC.CMS[1:0]) to set a count mode.

Count mode	Count mode selection bit (CMS[1:0])
To set the count mode to timer	Set the bit to "00".
To set the count mode to up/down count	Set the bit to "01".
To set the count mode to phase difference count (Multiply by 2)	Set the bit to "10".
To set the count mode to phase difference count mode (Multiply by 4)	Set the bit to "11".

41.7.3 How to select a count source for Up/Down Counter running in the timer mode

Use the internal prescaler select bit (UDCC.CLKS).

Count source for timer mode	Internal prescaler select bit (CLKS)
To obtain the CLKP divided by 2	Set the bit to "0".
To obtain the CLKP divided by 8	Set the bit to "1".

41.7.4 How to select the edge with which Up/Down Counter running in the Up/down count mode detects an input signal (AIN or BIN)

Use count clock edge select bits (UDCC.CES[1:0]).

Edge to be detected by counter	Count clock edge select bit (CES[1:0])
To disable detection	Set the bit to "00".
To enable detection of a falling edge	Set the bit to "01".
To enable detection of a rising edge	Set the bit to "10".
To enable detection of both edges	Set the bit to "11".

41.7.5 How to set a value to Up/Down Counter

A value can be set to Up/Down Counter by writing the value to the reload/compare register (RCR) and then writing "1" to the counter write bit (UDCC.CTUT).

41.7.6 When the Up/Down Counter's count-up value matches with the compare value (RCR[0:1]), how to enable clearing of Up/Down Counter the next time when the counter counts up

Use the up/down counter clear enable bit (UDCC.UCRE).

When the count-up value matches with the compare value and then Up/Down Counter counts up:	Up/down counter clear enable bit (UCRE)
To disable clearing of Up/Down Counter	Set the bit to "0".
To enable clearing of Up/Down Counter	Set the bit to "1".

MB91460 Series**41.7.7 How to enable reloading of the reload value (RCR[1:0]) to Up/Down Counter when Up/Down Counter is underflowed**

Use the reload enable bit (UDCC.RLDE).

When the count-up value matches with the compare value:	Reload enable bit (RLDE)
To disable reloading of the reload value (RCR) to Up/Down Counter	Set the bit to "0".
To enable reloading of the reload value (RCR) to Up/Down Counter	Set the bit to "1".

41.7.8 How to clear Up/Down Counter

Up/Down Counter can be cleared in any of the following ways:

- Writing "0" to the up/down counter clear bit (UDCC.UDCLR).
- Applying an edge to the ZIN pin (For details, refer to [41.7.9](#))
- When the compare value matches with the Up/Down Counter's count-up value.
- When Up/Down Counter tries to count up after reaching the maximum count.
- Reset input (INITX pin input, watchdog reset, software reset)

41.7.9 How to clear Up/Down Counter using the ZIN pin

Use counter clear gate bit (UDCC.CGSC) and counter clear gate edge select bits (UDCC.CGE[1:0]). (These bits are enabled in the up/down count mode.)

ZIN pin input	Counter clear gate bit (CGSC)	Counter clear gate edge select bit (CGE[1:0])
To disable edge detection (clear)	Set the bit to "0".	Set the bit to "00".
To clear Up/Down Counter on the falling edge	Set the bit to "0".	Set the bit to "01".
To clear Up/Down Counter on the rising edge	Set the bit to "0".	Set the bit to "10".

GCE[1:0]="11" indicates that setting is disabled.

41.7.10 How to control Up/Down Counter's count operation using the ZIN pin

Use counter clear gate bit (UDCC.CGSC) and counter clear gate edge select bits (UDCC.CGE[1:0]). (These settings are enabled for all the count modes.)

ZIN pin input	Counter clear gate bit (CGSC)	Counter clear gate edge select bit (CGE[1:0])
To disable level detection (counting)	Set the bit to "1".	Set the bit to "00".
To start counting up or down at the "L" level To stop counting up or down at the "H" level	Set the bit to "1".	Set the bit to "01".
To stop counting up or down at the "L" level To start counting up or down at the "H" level	Set the bit to "1".	Set the bit to "10".

GCE[1:0]= "11" indicates that setting is disabled.

41.7.11 How to enable/disable Up/Down Counter's count operation

Use the count activate bit (UDCS.CSTR).

When the count-up value matches with the compare value:	Count activate bits (UDCS.CSTR)
To disable Up/Down Counter's count operation	Set the bit to "0".
To enable Up/Down Counter's count operation (To activate count operation)	Set the bit to "1".

- How to start counting

Timer mode

Counting starts using the internal clock (See 41.7.3)

Up/down count mode

Counting starts when the edge of an AIN or BIN pin input signal is detected.
(See 41.7.4)

Phase difference count mode

Counting starts when a phase difference between AIN and BIN pins is detected.

Note that the count operation enable level must be detected, before the ZIN pin's gate function can be selected.

41.7.12 How to know the previous count direction (the current rotation direction)

Use the up/down flags (UDCS.UDF[1:0]).

Up/down flag (UDF[1:0])
"00" indicates that no counting is performed after resetting.
"01" indicates that counting down is performed.
"10" indicates that counting up is performed.
"11" indicates that both counting up and down are performed, resulting in no change in the count value.

This flag has nothing to do with interrupts. So, use the count direction change flag (UDCC.CDCF) for interrupt processing.

41.7.13 How to know count direction changes

Use the count direction change flag (UDCC.CDCF).

Count direction change flag (CDCF)
"0" indicates that no direction change has been made after clearing the flag.
"1" indicates that a direction change has been made once or more after clearing the flag.

41.7.14 How to know that a compare-match has occurred

Use the compare-match detection flag (UDCS.CMPF).

Compare-match detection flag (CMPF)
"0" indicates that the Up/Down Counter's count value does not match with the compare value.
"1" indicates that the Up/Down Counter's count value matches with the compare value.

Regardless of counter operations (counting up/down, or a value being set or reloaded), the compare-match detection flags are set to "1" when the count value matches with the compare value.

MB91460 Series

41.7.15 How to know that an overflow or underflow has occurred

Use the overflow detection flag (UDCS.OVFF) and the underflow detection flag (UDCS.UOFF).

OVFF = "1" indicates that Up/Down Counter has been overflowed.
UOFF = "1" indicates that Up/Down Counter has been underflowed.

41.7.16 How to set the reload/compare value

Set a value to the reload/compare registers (UDRC). (This value is used as a compare or reload value.)

41.7.17 What are interrupt-related registers?

Configure the up/down counter interrupt vectors and up/down counter interrupt level settings.

The following table shows the relationship among the up/down counter number, interrupt levels and vectors:

For details on interrupt levels and interrupt vectors, refer to "[Chapter 24 Interrupt Control \(Page No.429\)](#)".

	Interrupt vector (Default)	Interrupt level set bit (ICR[4:0])
Up/Down Counter 0/1 (16 bit)	#128 Address: 0FFDFCh	Interrupt level register (ICR56) Address: 0478h
Up/Down Counter 0 (8 bit)		
Up/Down Counter 1 (8 bit)	#129 Address: 0FFDF8h	
Up/Down Counter 2/3 (16 bit)	#130 Address: 0FFDF4h	Interrupt level register (ICR57) Address: 0479h
Up/Down Counter 2 (8 bit)		
Up/Down Counter 3 (8 bit)	#131 Address: 0FFDF0h	

The following interrupt flags are not automatically cleared:

- Count direction change: UDCC.CDCF
- Compare-match detection: UDCS.CMPF
- Overflow: UDCS.OVFF
- Underflow: UDCS.UOFF

So, the software must write "0" to the interrupt flag before control is returned from interrupt processing.

41.7.18 What interrupts are available and how are they selected?

There are four interrupt causes:

- Count direction change
- compare-match
- overflow
- underflow

An interrupt request is made by combining (logical OR) these four interrupt causes; each interrupt cause cannot be isolated.

Use the interrupt request enable bit to enable a desired interrupt request.

41.7.19 How to enable (select), disable or clear interrupts

Interrupt request enable bits and interrupt flags

To enable (select) interrupts, use the following interrupt request enable bits:

- Count direction change interrupt request enable bits : UDCC.CFIE
- Compare-match interrupt request enable bits : UDCC.CITE
- Overflow/underflow interrupt request enable bits : UDCC.UDIE

	Interrupt request enable bits (CFIE, CITE and UDIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupts, use the following interrupt flags:

- For count direction changes : UDCC.CDCF
- For compare-match detection : UDCC.CMPF
- For overflow : UDCC.OVFF
- For underflow : UDCC.UDFF

	Interrupt flags (CDCF, CMPF, OVFF and UDFF)
To clear terrupts	Write "0".

41.8. Caution

- The count direction is set to “countdown” immediately after resetting the counter. So, when the counter counts up immediately after resetting, the count direction change bit (UDCC.CDCF) is set to “1” to indicate a direction change has been made.
- When the up/down counter register UDCR has reached the maximum count, the overflow flag is set to 1 and counting continues. This time UDCR is cleared.
- The minimum pulse width for AIN, BIN and ZIN signals is $2 \times T$ ($T = 1/\text{CLKP}$: Period of a peripheral clock)
- If it is determined whether a change has been made to the count direction change interrupt and count direction flags, it must be taken into consideration that when several direction changes have been made continuously in a short period of time, the count direction flag may be returned to the original value, which looks as if no change has been made.
- The compare-match detection flag (UDCS.CMPF) is set to “1” when Up/Down Counter's count value matches with the compare value during both counting up and down. These flags are also set to 1 when:
 - The reload value is reloaded to Up/Down Counter; or
 - The Up/Down Counter's count value matches with the compare value when Up/Down Counter is activated.
- The Up/Down Counter's count value is cleared by a clear request which is generated:
 - On the edge of a signal input from the ZIN pin;
 - By writing “0” to the up/down counter clear bit (UDCC.UDCLR); or
 - When the compare value matches with the count value.
 In addition,
 - On reset input (INITX pin input, RST and watchdog reset); or
 - When the counter counts up from the maximum account, the count value is also set to “0000_H”.
- When Up/Down Counter clear and reload requests are made at the same time, the clear operation has higher priority.
- When Up/Down Counter is counting up, writing to the counter is disabled.
Writing “1” to the counter write bit (UDCC.CTUT) after writing to the UDRC register is disabled. If writing to the reload register and a reload operation occurs at the same time, the writing to the reload register is not performed.
- The software cannot clear the up/down flags (UDCS.UDF[1:0]) to “0”. Only reset (initialization) can clear the flag to “0”.

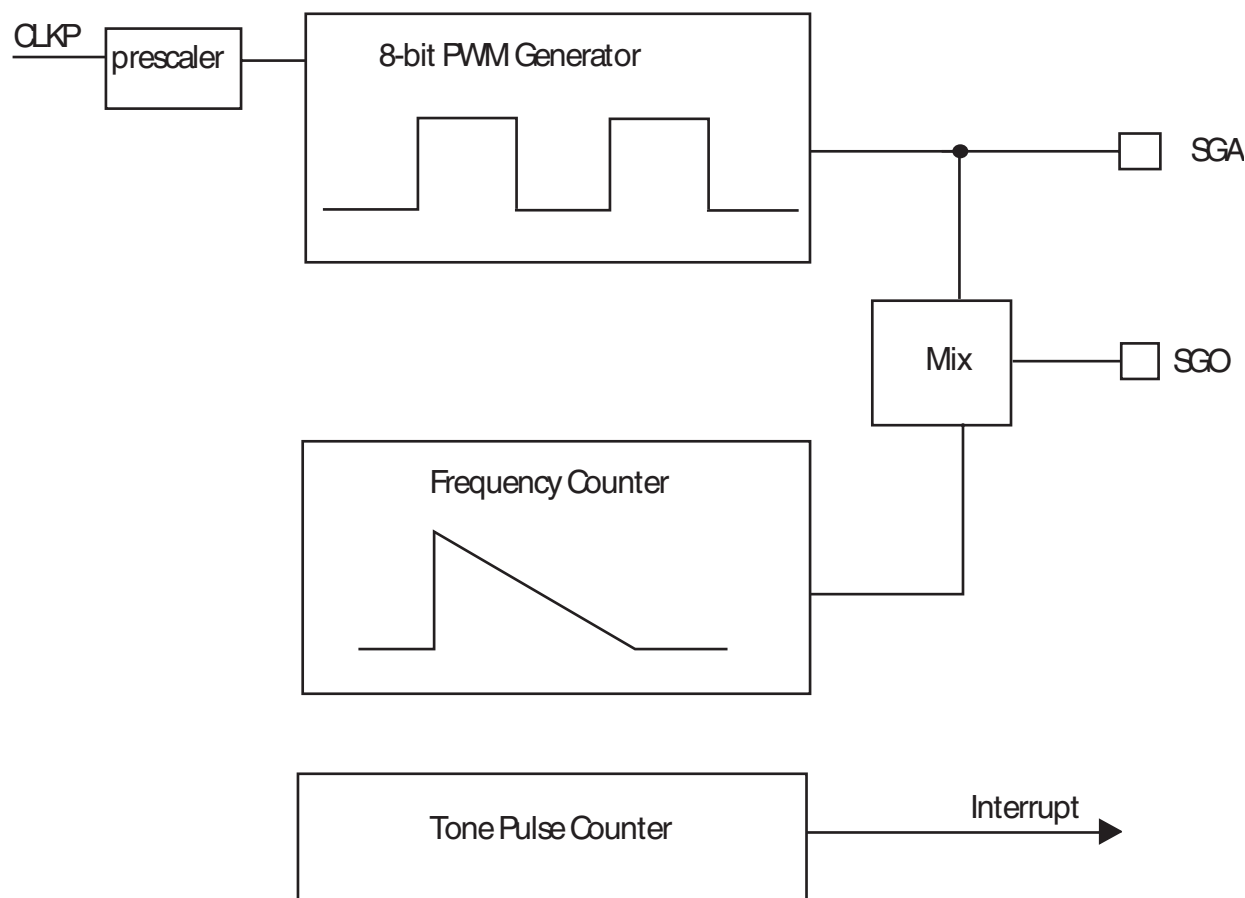
Chapter 42 Sound Generator (SG)

42.1. Overview

This Chapter provides an overview of the Sound Generator, describes the register structure and functions, and describes the operation of the Sound Generator.

The Sound Generator consists of the Sound Control register, Frequency Data register, Amplitude Data register, Decrement Grade register, Tone Count register, Sound Disable register, PWM generator, Frequency counter, Decrement counter and Tone Pulse counter.

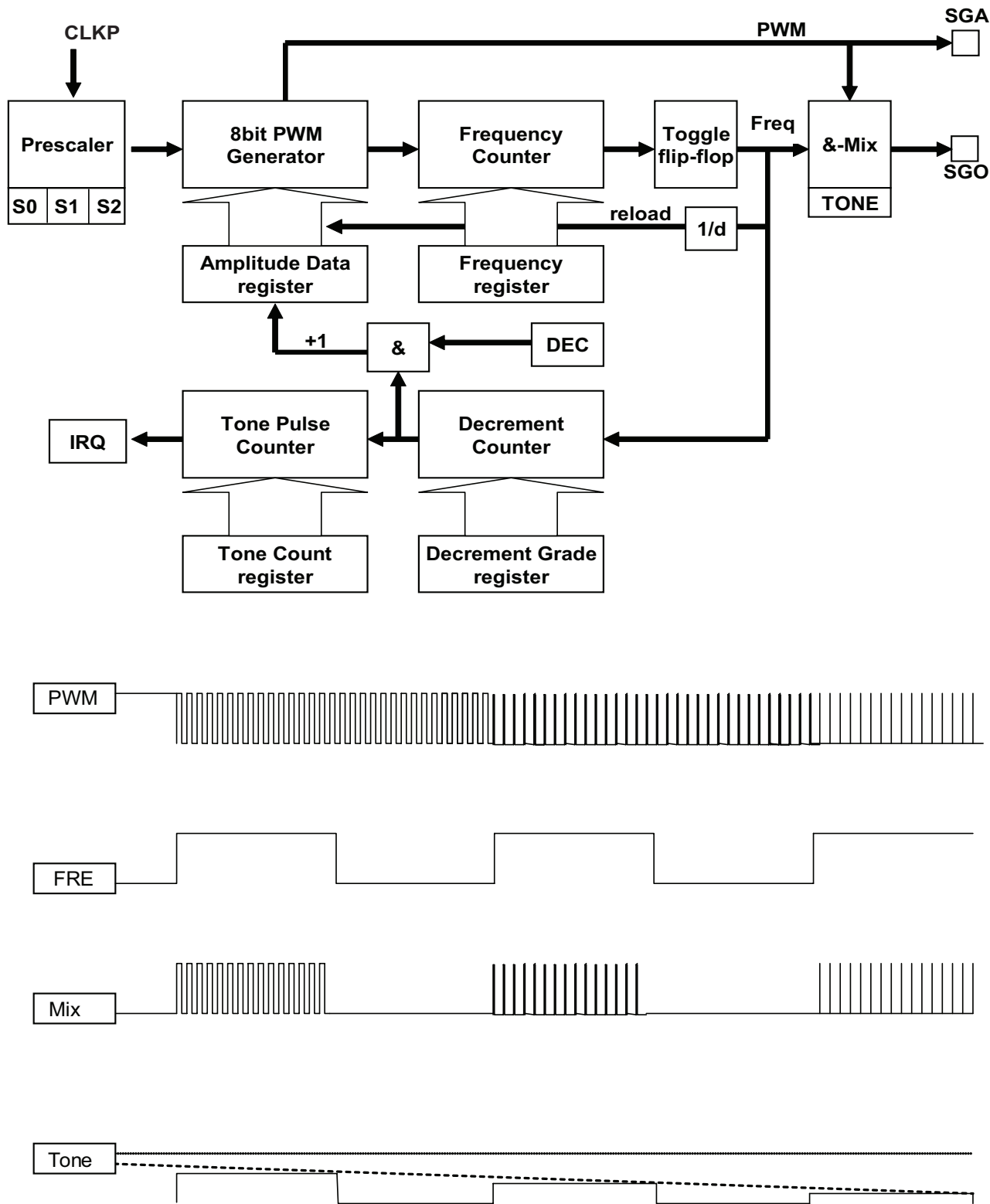
Figure 42.1-1 Block diagram of the sound generator



Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

42.2. Detailed Structure of the Sound generator

Figure 42.2-1 Detailed Structure of the sound generator



MB91460 Series

42.3. Registers

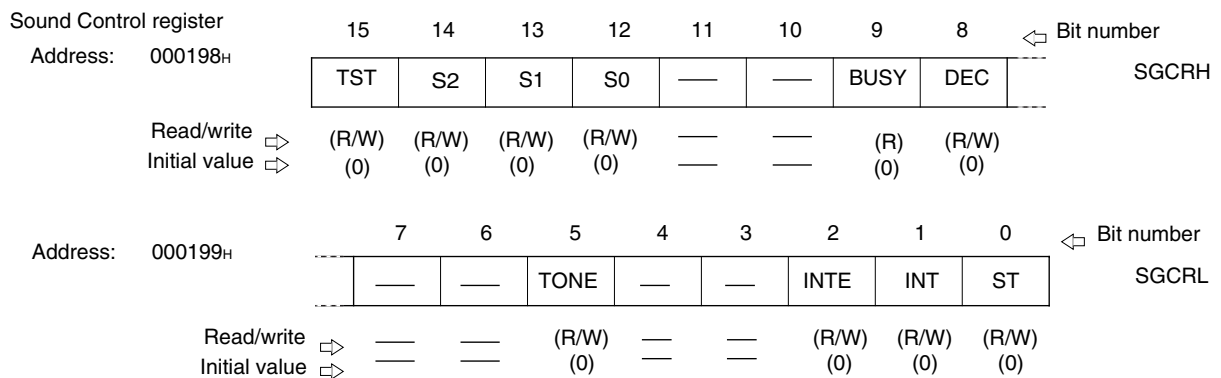
Figure 42.3-1 Sound Generator Register overview

Sound Control register		15	14	13	12	11	10	9	8	Bit number
Address:	000198 _H	TST	S2	S1	S0	—	—	BUSY	DEC	SGCRH
Read/write	⇒ (R/W)	(R/W)	(R/W)	(R/W)	—	—	(R)	(R/W)		
Initial value	⇒ (0)	(0)	(0)	(0)	—	—	(0)	(0)		
Frequency Data register		7	6	5	4	3	2	1	0	Bit number
Address:	000199 _H	—	—	TONE	—	—	INTE	INT	ST	SGCRL
Read/write	⇒ —	—	(R/W)	—	—	(R/W)	(R/W)	(R/W)		
Initial value	⇒ —	—	(0)	—	—	(0)	(0)	(0)		
Amplitude Data register		15	14	13	—	3	2	1	0	Bit number
Address:	00019A _H	D15	D14	D13	—	D3	D2	D1	D0	SGFR
Read/write	⇒ (R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	⇒ (X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
Device Auto Disable register		7	6	5	4	3	2	1	0	Bit number
Address:	00019D _H	—	—	—	—	—	—	SDADS	SDADR	SGDAD
Read/write	⇒ —	—	—	—	—	—	—	(R)	(R/W)	
Initial value	⇒ —	—	—	—	—	—	—	(0)	(0)	
Tone Count register		15	14	13	12	11	10	9	8	Bit number
Address:	00019E _H	D7	D6	D5	D4	D3	D2	D1	D0	SGTR
Read/write	⇒ (R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	⇒ (X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
Decrement Grade register		7	6	5	4	3	2	1	0	Bit number
Address:	00019F _H	D7	D6	D5	D4	D3	D2	D1	D0	SGDR
Read/write	⇒ (R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	⇒ (X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Note: The Device Auto Disable register SGDAD is implemented on MB91FV460B and MB91F467E only.

42.3.1 Sound Control Register (SGCR)

Figure 42.3-2 Sound Control Register (SGCR)



[bit 15] TST : Test bit

This bit is reserved for the device test. In every user application it should be set to "0".

[bits14 to12] S2 to S0 : Operation clock select bits

These bits specify the clock input signal for the Sound Generator.

Table 42.3-1 Sound generator clock selection

S2	S1	S0	Clock input
0	0	0	CLKP
0	0	1	CLKP/2
0	1	0	CLKP/4
0	1	1	CLKP/8
1	x	x	CLKP/16

[bit 9] BUSY : Busy bit

This bit indicates whether the Sound Generator is in operation. This bit is set to "1" after the ST bit is set to "1". It is reset to "0" when the ST bit is reset to "0" and the operation is completed at the end of one tone period. Any write instruction performed on this bit has no effect.

[bit 8] DEC : Auto-decrement enable bit

The DEC bit is used for an automatic decrement of the sound. This automatic decrement is configured with the Decrement Grade register.

If this bit is set to "1", the stored value in the Amplitude Data register is decremented by 1(one), every time when the Decrement counter counts the number of tone pulses from the toggle flip-flop specified by the Decrement Grade register.

[bit 5] TONE : Tone output bit

When this bit is set to "1", the SGO signal becomes a simple square-waveform (tone pulses) from the toggle flip-flop. Otherwise it is the mixed (logical AND) signal of the tone and PWM

pulses.

[bit 2] INTE : Interrupt request enable bit

This bit enables the interrupt request of the Sound Generator. When this bit is "1" and the INT bit is set to "1", the Sound Generator generates an interrupt request.

[bit 1] INT : Interrupt flag

This flag is set to "1" after the Tone Pulse counter has counted the number of the tone pulses specified by the Tone Count register and Decrement Grade register.

This bit is reset to "0" by writing "0". Writing "1" has no effect and Read-Modify-Write instructions always result in reading "1".

[bit 0] ST : Start bit

This bit is for starting the operation of the Sound Generator. While this bit is "1", the Sound Generator performs its operation.

When this bit is reset to "0", the Sound Generator stops its operation at the end of the current tone period. The BUSY bit indicates whether the Sound Generator is fully stopped.

42.3.2 Frequency Data Register (SGFR)

Figure 42.3-3 Frequency Data Register (SGFR)

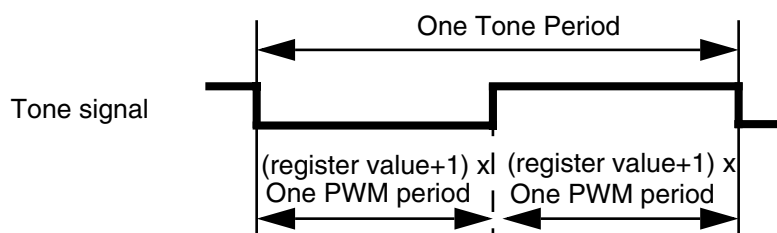
Frequency Data register								
Address: 00019AH								
	15	14	13		3	2	1	0
	D15	D14	D13		D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)		(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)		(X)	(X)	(X)	(X)

Bit number
SGFR

The Frequency Data register stores the reload value for the Frequency counter. The stored value represents the frequency of the sound (or the tone signal from the toggle flip-flop). The register value is reloaded into the counter at every transition of the toggle signal.

The following figure shows the relationship between the tone signal and the register value.

Figure 42.3-4 Tone period of Sound Generator



It should be noted that modifications of the register value while operation may alter the duty cycle of 50% depending on the timing of the modification.

42.3.3 Amplitude Data Register (SAGR)

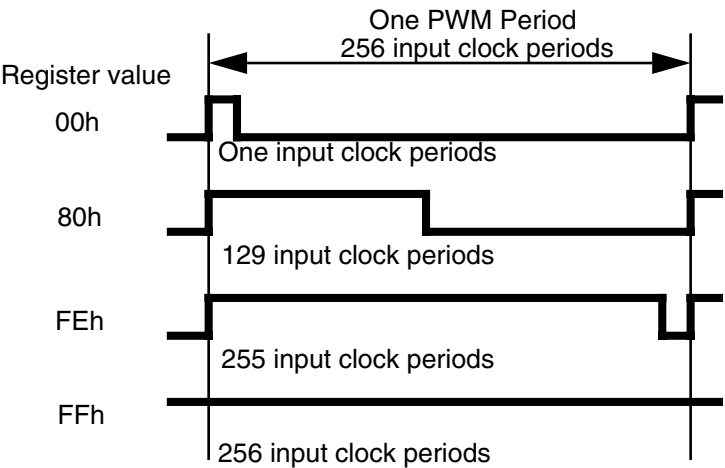
Figure 42.3-5 Amplitude Data Register (SAGR)

Amplitude Data register	15	14	13	12	11	10	9	8	Bit number
Address: 00019C _H	D7	D6	D5	D4	D3	D2	D1	D0	SGAR
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

The Amplitude Data register stores the reload value for the PWM generator. The register value represents the amplitude of the sound. The register value is reloaded into the PWM generator at the end of every tone period.

When the DEC bit is "1" and the Decrement counter reaches its reload value, this register value is decremented by 1(one). And when the register value reaches "00", further decrements are not performed. However the sound generator continues its operation until the ST bit is cleared.

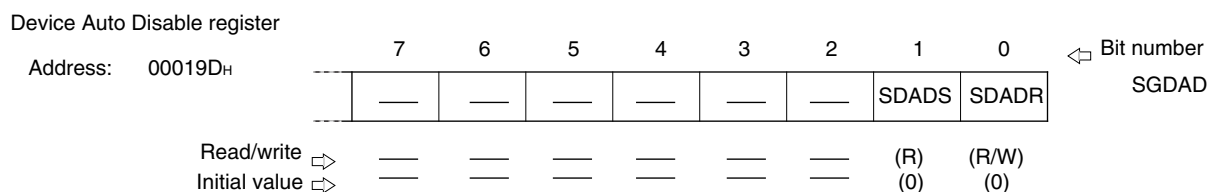
Figure 42.3-6 Relationship between the register value and the PWM pulse



When the register value is set to "FF", the PWM signal is always "1".

42.3.4 Device Auto Disable Register (SGDAD)

Figure 42.3-7 Device auto disable register (SGDAD)



The device auto disable register is implemented on MB91FV460B and MB91F467E only.

This register controls the total disable of SGA, SGO output when the amplitude is 00_H.

[bits 7:2] : unused bits

These bits are not used. Read operation returns '0'.

[bit 1] SDADS : Auto Disable Status flag

This read-only flag is set to "1" after the outputs SGA, SGO have been disabled by the auto disable feature.

This bit is cleared by reset (RST) or when SDADR is cleared or when the amplitude register is not equal 00_H.

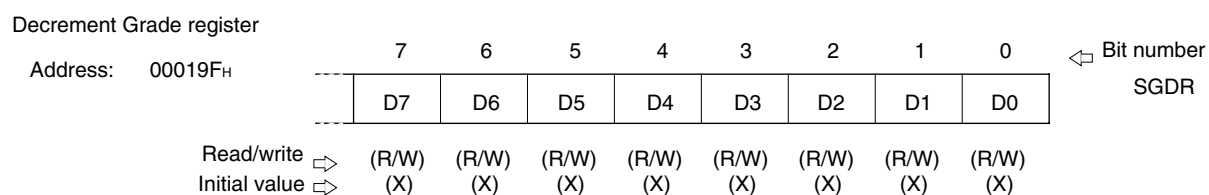
[bit 0] SDADR : Auto Disable enable bit

This bit enables the auto disable feature. If this bit is '1' and the amplitude is 00_H, the outputs SGA, SGO are switched off.

This bit is cleared by reset (RST) or by writing '0'.

42.3.5 Decrement Grade Register (SGDR)

Figure 42.3-8 Decrement Grade Register (SGDR)



The Decrement Grade register stores the reload value for the Decrement counter. They are prepared to automatically decrement the stored value in the Amplitude Data register.

When the DEC bit is "1" and the Decrement counter counts the number of tone pulses up to the reload value, the stored value in the Amplitude Data register is decremented by 1 (one) at the end of the tone period.

This operation realizes automatic decrement of the sound with fewer number of CPU interventions.

It should be noted that the number of the tone pulses specified by this register equals to "register value +1". When the Decrement Grade register is set to "00", the decrement operation is performed every tone period.

42.3.6 Tone Count Register (SGTR)

Figure 42.3-9 Tone Count Register (SGTR)

Tone Count register		15	14	13	12	11	10	9	8	Bit number
Address: 00019EH										
		D7	D6	D5	D4	D3	D2	D1	D0	SGTR
Read/write	⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

The Tone Count register stores the reload value for the Tone Pulse counter. The Tone Pulse counter accumulates the number of tone pulses (or number of decrement operations) and when it reaches the reload value it sets the INT bit. They are intended to reduce the frequency of interrupts.

The count input of the Tone Pulse counter is connected to the carry-out signal from the Decrement counter. And when the Tone count register is set to "00", the Tone Pulse counter sets the INT bit every carry-out from the Decrement counter. Thus the number of accumulated tone pulses is:
 $((\text{Decrement Grade register}) + 1) \times ((\text{Tone Count register}) + 1)$

i.e. When the both registers are set to "00", the INT bit is set at every tone period.

Chapter 43 Stepper Motor Controller (SMC)

43.1. Overview

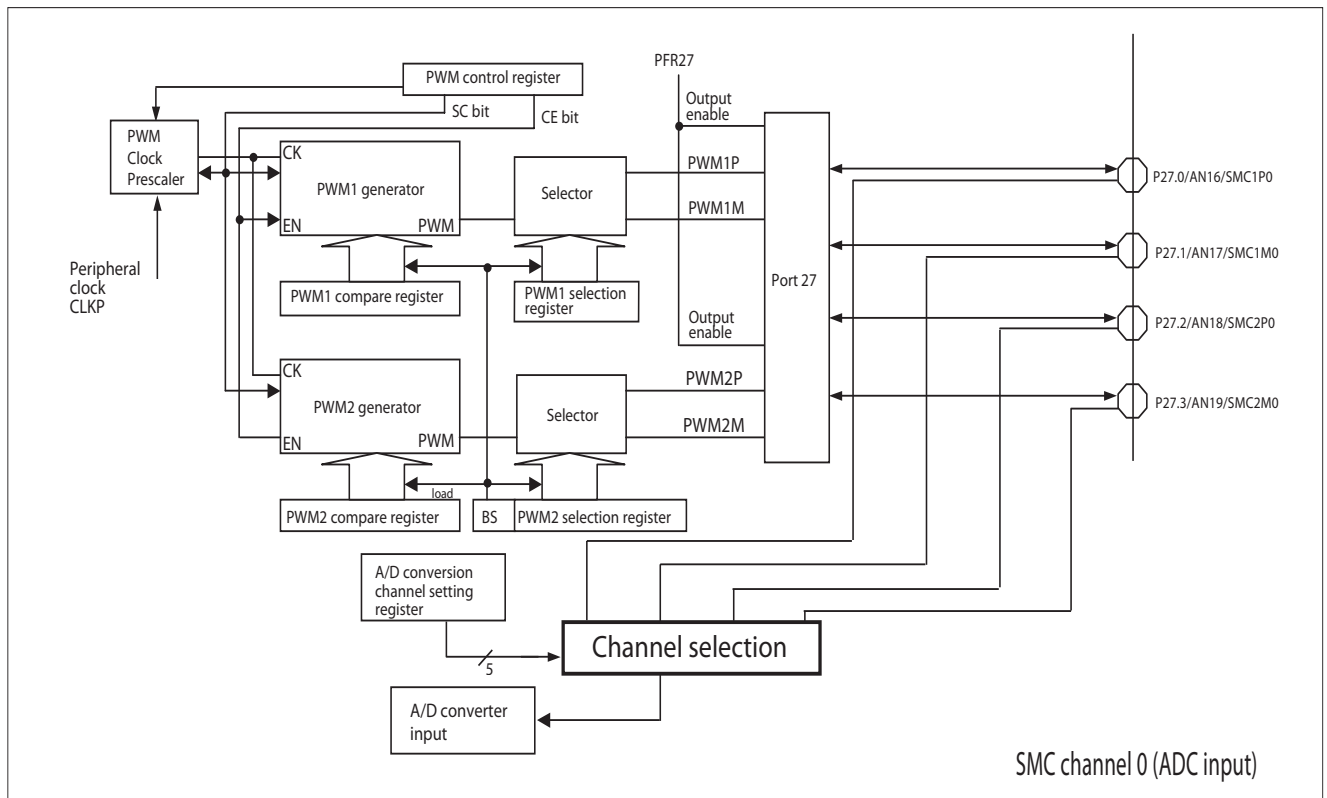
The Stepper Motor Controller consists of PWM generators, motor drivers, selectors and A/D converter inputs. The ADC inputs can be used to measure the voltage level at the SMC pins during operation.

The SMCs have high current output drivers, where two motor coils can be connected directly (No further driver is necessary). The combination of the PWM generators and selector logic circuits is designed to control the motor rotation. The synchronization mechanism enables synchronous operation of two PWM generators.

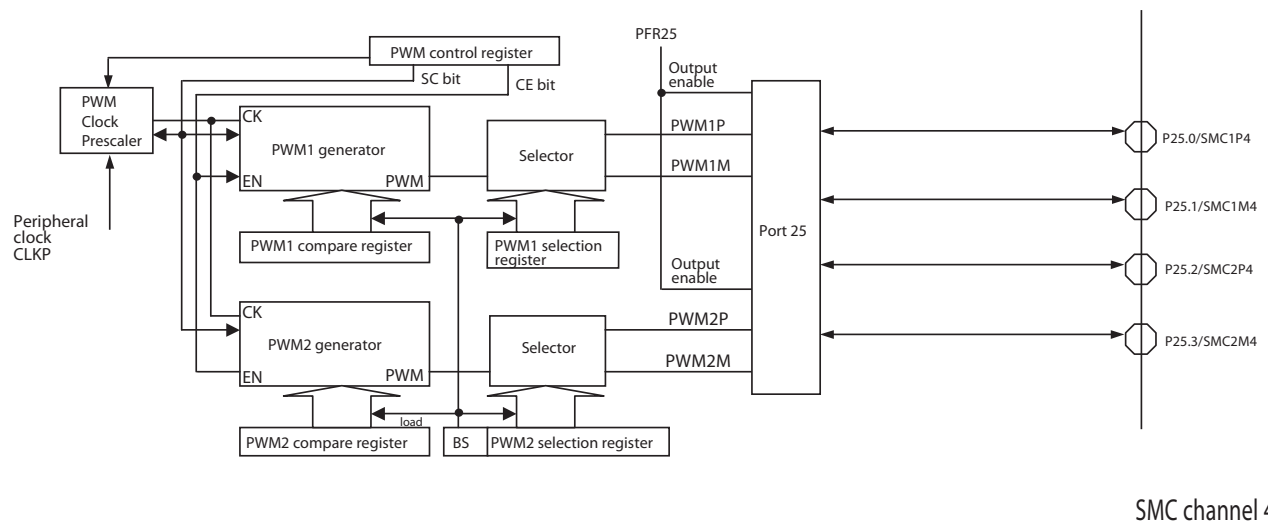
Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

43.1.1 Block Diagram of Stepper Motor Controller

Figure 43.1-1 Stepper Motor Controller with A/D Converter Function



Note: The SMC channels 0, 1, 2 and 3 are shared with ADC inputs.

Figure 43.1-2 Stepper Motor Controller

Note: The SMC channels 4 and 5 are not shared with ADC inputs.

43.2. Registers

There are seven types of registers for the stepper motor controller:

- PWM Control register
- PWM1 Compare register
- PWM2 Compare register
- PWM1 Selection register
- PWM2 Selection register

43.2.1 List of the Stepper Motor Controller Registers

Figure 43.2-1 Stepper Motor Controller Register overview

PWM Control register (PWC0, PWC1, PWC2, PWC3, PWC4, PWC5)								
Address	7	6	5	4	3	2	1	0
0x0C1, 0x0C3	–	P2	P1	P0	CE	SC	–	–
0x0C5, 0x0C7	–	R/W	R/W	R/W	R/W	R/W	–	–
0x0C9, 0x0CB	–	0	0	0	0	0	–	–
PWM1 Compare register (PWC10, PWC11, PWC12, PWC13, PWC14, PWC15)								
Address	15	14	13	12	11	10	9	8
0x092, 0x09A	–	–	–	–	–	–	D9	D8
0x0A2, 0x0AA	–	–	–	–	–	–	R/W	R/W
0x0B2, 0x0BA	–	–	–	–	–	–	X	X
Address	7	6	5	4	3	2	1	0
0x093, 0x09B	D7	D6	D5	D4	D3	D2	D1	D0
0x0A3, 0x0AB	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0B3, 0x0BB	X	X	X	X	X	X	X	X
PWM2 Compare register (PWC20, PWC21, PWC22, PWC23, PWC24, PWC25)								
Address	15	14	13	12	11	10	9	8
0x090, 0x098	–	–	–	–	–	–	D9	D8
0x0A0, 0x0A8	–	–	–	–	–	–	R/W	R/W
0x0B0, 0x0B8	–	–	–	–	–	–	X	X
Address	7	6	5	4	3	2	1	0
0x091, 0x099	D7	D6	D5	D4	D3	D2	D1	D0
0x0A1, 0x0A9	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0B1, 0x0B9	X	X	X	X	X	X	X	X
PWM2 Selection register (PWS20, PWS21, PWS22, PWS23, PWS24, PWS25)								
Address	15	14	13	12	11	10	9	8
0x096, 0x09E	–	BS	P2	P1	P0	M2	M1	MO
0x0A6, 0x0AE	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0B6, 0x0BE	–	0	0	0	0	0	0	0
PWM1 Selection register (PWS10, PWS11, PWS12, PWS13, PWS14, PWS15)								
Address	7	6	5	4	3	2	1	0
0x097, 0x09F	–	–	P2	P1	P0	M2	M1	MO
0x0A7, 0x0AF	–	–	R/W	R/W	R/W	R/W	R/W	R/W
0x0B7, 0x0BF	–	–	0	0	0	0	0	0

43.2.2 PWM Control Register

The PWM control register enables/disables the stepper motor controller, sets the clock prescaler and determines whether the PWM generator operates at 8 bit or 10 bit.

■ PWM Control Register

Figure 43.2-2 PWM Control Register

PWM Control register (PWC0, PWC1, PWC2, PWC3, PWC4, PWC5)

Address	7	6	5	4	3	2	1	0
0x0C1, 0x0C3	–	P2	P1	P0	CE	SC	–	–
0x0C5, 0x0C7	–	R/W	R/W	R/W	R/W	R/W	–	–
0x0C9, 0x0CB	–	0	0	0	0	0	–	–

[bit 7] Reserved bit

Always set the reserved bit to "0".

[bit 6 to 4] P2, P1, P0: Operating clock select bits (bits to select operating clock)

The P2, P1 and P0 bits set the Clock prescaler to specify the input signal frequency for the PWM generator.

Table 43.2-1 SMC clock selection

P2	P1	P0	Clock input	PWM period (at CLKP = 32MHz)		PWM period (at CLKP = 40MHz)	
				SC=0	SC=1	SC=0	SC=1
0	0	0	CLKP	8.0 us	32.0 us	6.4 us	25.6 us
0	0	1	CLKP/4	32.0 us	128.0 us	25.6 us	102.4 us
0	1	0	CLKP/5	40.0 us	160.0 us	32.0 us	128.0 us
0	1	1	CLKP/6	48.0 us	192.0 us	38.4 us	153.6 us
1	0	0	CLKP/8	64.0 us	256.0 us	51.2 us	204.8 us
1	0	1	CLKP/10	80.0 us	320.0 us	64.0 us	256.0 us
1	1	0	CLKP/12	96.0 us	384.0 us	76.8 us	307.2 us
1	1	1	CLKP/16	128.0 us	512.0 us	102.4 us	409.6 us

Note: After operation clock selection, set 1 in CE.

[bit 3] CE: Count enable bit

The CE bit enables the operation of the PWM generator. When "1" is set to the CE bit, the PWM generator starts its operation. The PWM2 generator starts after the PWM1 generator in order to reduce the switching noise generated by the output driver.

When the CE bit is cleared to 0 during operation of the PWM generator, the generator stops operating.

[bit 2] SC: 8/10 bits switching bit

When "1" is set to the SC bit, the PWM generator operates at 10 bit. When "0" is set to the SC bit, the PWM generator operates at 8 bit.

[bit 1 to 0] Reserved bits

Always set the reserved bits to "00".

MB91460 Series

43.2.3 PWM1 and PWM2 Compare Registers

Depending on the value of the SC bit the bit length of the PWM Compare Register is either 8 or 10 bits.

The value stored in the PWM Compare Register determines the PWM pulse width. For example, a value equal to 00_H (000_H) set the duty cycle to 0% and a value equal to FF_H (3FF_H) set the duty cycle to 99.6% (99.9%).

■ PWM1 and PWM2 Compare Registers

The PWM1 and PWM2 compare registers can be accessed at any time, but the changed value is reflected in the pulse width at the end of the current PWM period after "1" is set to the BS bit of the PWM2 selection register.

If the SC bit is set to "0", the PWM generator performs a 8-bit operation and the bits D9-D8 are undefined.

It is necessary to perform half-word access to the PWM1 and PWM2 compare registers

Figure 43.2-3 PWM1 and PWM2 Compare Registers

PWM1 Compare register (PWC10, PWC11, PWC12, PWC13, PWC14, PWC15)							
Address	15	14	13	12	11	10	9 8
0x092, 0x09A	—	—	—	—	—	—	D9 D8
0x0A2, 0x0AA	—	—	—	—	—	—	R/W R/W
0x0B2, 0x0BA	—	—	—	—	—	—	X X
Address	7	6	5	4	3	2	1 0
0x093, 0x09B	D7	D6	D5	D4	D3	D2	D1 D0
0x0A3, 0x0AB	R/W	R/W	R/W	R/W	R/W	R/W	R/W R/W
0x0B3, 0x0BB	X	X	X	X	X	X	X X
PWM2 Compare register (PWC20, PWC21, PWC22, PWC23, PWC24, PWC25)							
Address	15	14	13	12	11	10	9 8
0x090, 0x098	—	—	—	—	—	—	D9 D8
0x0A0, 0x0A8	—	—	—	—	—	—	R/W R/W
0x0B0, 0x0B8	—	—	—	—	—	—	X X
Address	7	6	5	4	3	2	1 0
0x091, 0x099	D7	D6	D5	D4	D3	D2	D1 D0
0x0A1, 0x0A9	R/W	R/W	R/W	R/W	R/W	R/W	R/W R/W
0x0B1, 0x0B9	X	X	X	X	X	X	X X

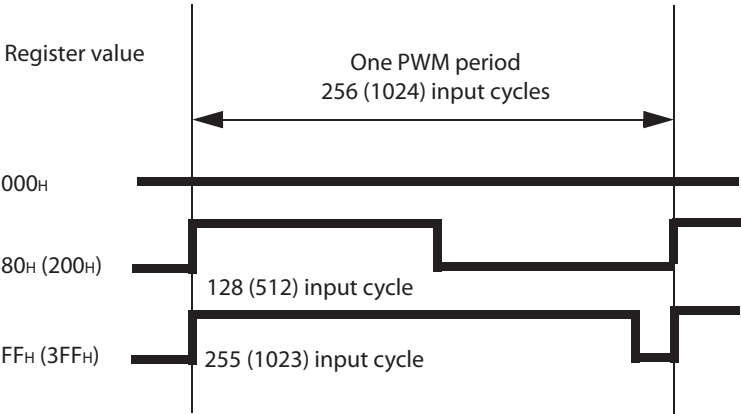
[bit 15 to 10] Reserved bit

Always set the reserved bit to "0".

[bits 9 to 0] D9 to D0: Compare data

These bits are used to set the PWM pulse width.

Figure 43.2-4 Relationship between the Compare Register Setting Value and PWM Pulse Width



MB91460 Series

43.2.4 PWM1 and PWM2 Selection Registers

The PWM Selection Registers PWM1 and PWM2 select the signal on the output pin of the stepper motor controller. It is possible to select among low level, high level, PWM signal and high impedance. In addition, these register contain a control bit.

■ PWM1 and PWM2 Selection Registers

Figure 43.2-5 PWM1 and PWM2 Selection Registers

PWM2 Selection register (PWS20, PWS21, PWS22, PWS23, PWS24, PWS25)								
Address	15	14	13	12	11	10	9	8
0x096, 0x09E	–	BS	P2	P1	P0	M2	M1	M0
0x0A6, 0x0AE	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0x0B6, 0x0BE	–	0	0	0	0	0	0	0
PWM1 Selection register (PWS10, PWS11, PWS12, PWS13, PWS14, PWS15)								
Address	7	6	5	4	3	2	1	0
0x097, 0x09F	–	–	P2	P1	P0	M2	M1	M0
0x0A7, 0x0AF	–	–	R/W	R/W	R/W	R/W	R/W	R/W
0x0B7, 0x0BF	–	–	0	0	0	0	0	0

[bit 15] Reserved bit

Always set the reserved bit to "0".

[bit 14] BS: Update bit

This bit is prepared to synchronise the settings for the PWM outputs. Any modifications in the two compare registers and two select registers are not reflected to the output signals until this bit is set.

When this bit is set to "1", the PWM generators and selectors load the register contents at the end of the current PWM cycle. The BS bit is reset to "0" automatically at the beginning of the next PWM cycle. If the BS bit is set to "1" by software at the same time as this automatic reset, the BS bit is set to "1" (or remains unchanged) and the automatic reset is cancelled.

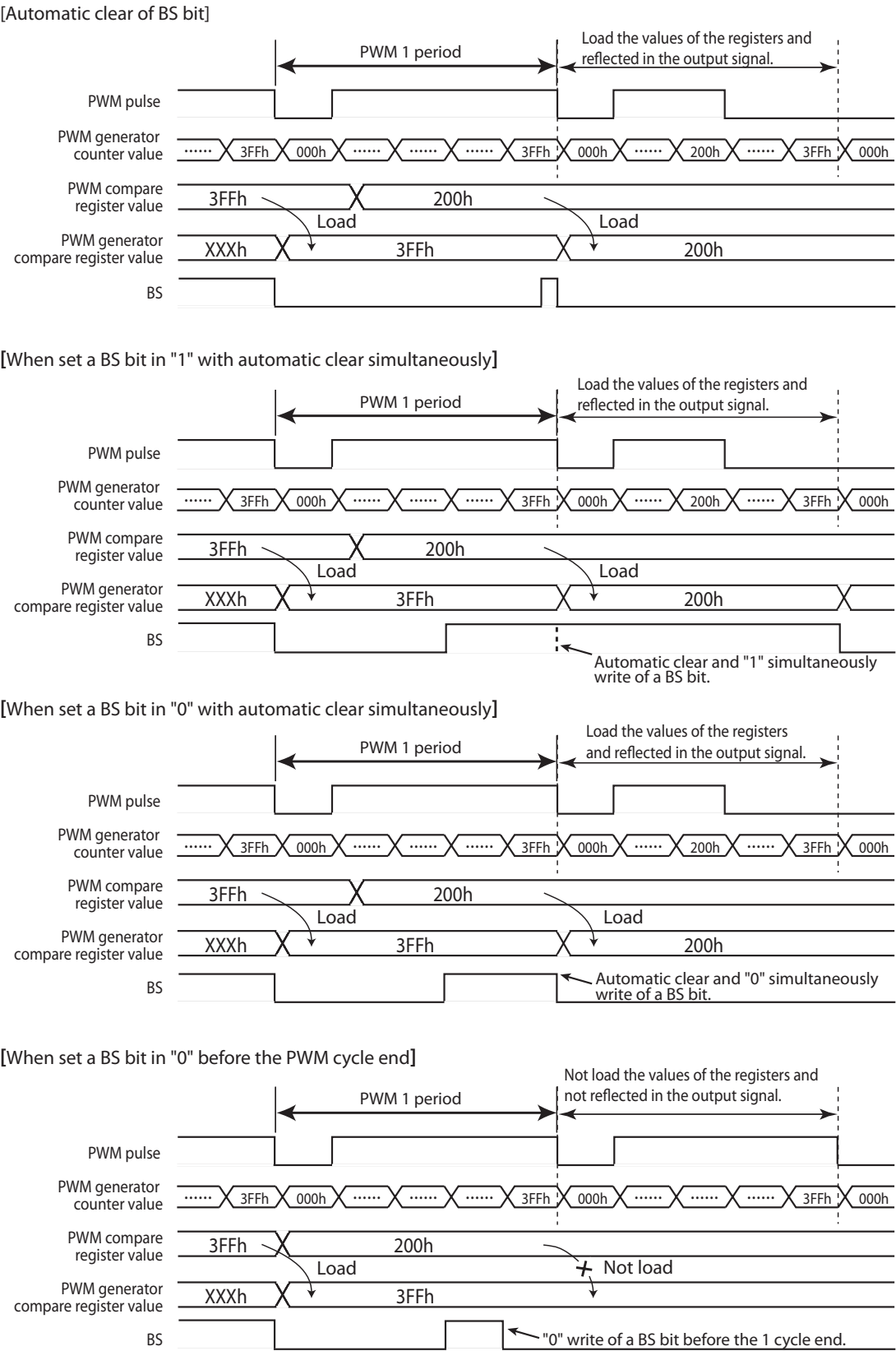
Note:

When the BS bit equals "1" when executing a read-modify-write type instruction the BS bit is "1" read as "1" and wrote as "1" in the BS bit again.

When the BS bit was cleared automatically by the starting of a PWM period in between read and write, "1" is set after BS bit has cleared again.

Therefore, values of the registers are loaded in the PWM generator and selector when the BS bit is not set to "1" by the end of the next PWM period.

Figure 43.2-6 load timing of PWM compare register value



MB91460 Series**[bit 13 to 11] P2 to P0: Output select bits**

These bits are used to select the output signal for SMC2P.

[bit 10 to 8] M2 to M0: Output select bits

These bits are used to select the output signal for SMC2M.

[bit 7 to 6] Reserved bit

Always set the reserved bit to "0".

[bit 5 to 3] P2 to P0: Output select bits

These bits are used to select the output signal for SMC1P.

[bit 2 to 0] M2 to M0: Output select bits

These bits are used to select the output signal for SMC1M.

The table below shows the relationship between the output levels and the select bits.

Table 43.2-2 SMC output level selection

P2	P1	P0	PWMmPn	M2	M1	M0	PWMmMn
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWM Pulse	0	1	X	PWM Pulse
1	X	X	High impedance	1	X	X	High impedance

m = 1 to 2 (motor coils)

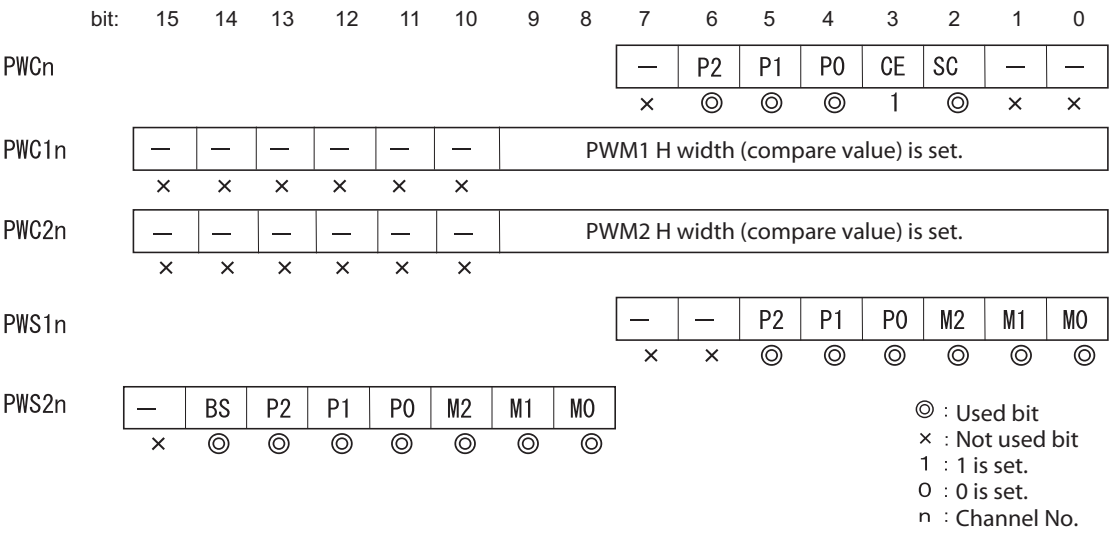
n = 0 to 5 (stepper motor channels)

43.3. Operation

The operation of the stepper motor controller is explained in this section.

43.3.1 Setting Operation of Stepper Motor Controller

Figure 43.3-1 Setting of Stepper Motor Controller



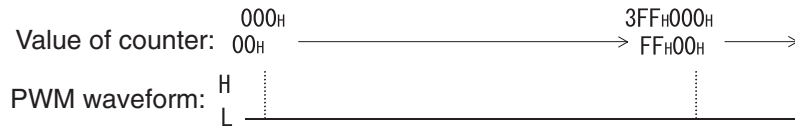
43.3.2 Operation of PWM generator

When the counter is started (PWC: CE = 1), it starts incrementing from 00_H on each rising edge from the prescaled count clock. The PWM output remains "H" until the value of the counter matches the value set to PWM compare register, and then changes to "L" and remains so until the counter overflows (FF_H --> 00_H).

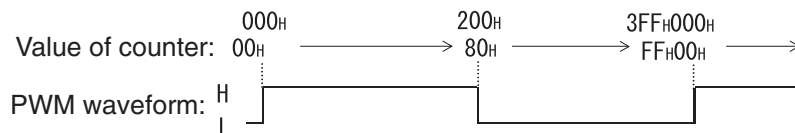
Figure 43.3-2 "Examples of PWM1 and PWM2 Waveform Output" shows the PWM waveform generated by the PWM generator

Figure 43.3-2 Examples of PWM1 and PWM2 Waveform Output

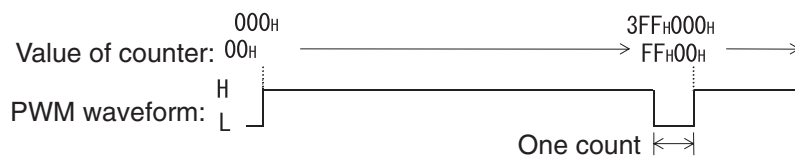
When the value of compare register is "00H"/"000H"(duty ratio is 0%):



When the value of compare register is "80H"/"200H"(duty ratio is 50%):



When the value of compare register is "FFH"/"3FFH"(duty ratio is 99.6%/99.9%):



43.3.3 Selection of motor drive signals

Motor drive signals that are output to each stepper motor controller pin can be selected among four types of signals for each pin by setting the PWM selection register.

Table 43.3-1 "Selection of Motor Drive Signals and Setting of PWM Selection Registers PWM1 and PWM2" lists the selection of the motor drive signals and the settings of PWM1 and PWM2 selection registers.

When these registers are set and "1" is written to the BS bit of the PWM selection register 2, the setting of these registers is enabled at the end of the current PWM period. The BS bit is cleared automatically to 0 at the beginning of the next PWM period. When "1" is written to the BS bit, and the BS bit is cleared to 0 simultaneously at the beginning of the next PWM period, "1" is written to the BS bit and clearing of the BS bit is cancelled.

Table 43.3-1 Selection of Motor Drive Signals and Setting of PWM1 and PWM2 Selection Registers

P2, P1, P0 Bits	PWM2P Output PWM1P Output	M2, M1, M0 Bits	PWM1M Output PWM2M Output
000 _B	L	000 _B	L
001 _B	H	001 _B	H
01X _B	PWM Pulse	01X _B	PWM Pulse
1XX _B	High impedance	1XX _B	High impedance

43.4. Caution

The caution when using the stepper motor controller are described below.

43.4.1 Caution when changing PWM Setting

The PWM1 and PWM2 compare registers (PWC1, PWC2) and the PWM1 and PWM2 selection registers (PWS1, PWS2) can be accessed at any time. However, to change the setting of the "H" width from the PWM or to change the PWM output, "1" must be written to the BS bit of the PWM2 selection register after (or at the same time) a setting is written to those registers (the PWM1 and PWM2 compare register and the PWM1 and PWM2 selection register).

When "1" is set to the BS bit, the new setting is enabled at the end of the current PWM period and the BS bit is cleared automatically.

Also, when "1" is written to the BS bit and the BS bit is reset at the end of the PWM period simultaneously, "1" is written to the BS and resetting of the BS bit is cancelled.

Chapter 44 A/D Converter (ADC) / Range Comparator

MB91460 series devices have slightly different A/D converter macros and different numbers of channels. Some devices are equipped with a digital Range Comparator. This chapter describes the A/D converters, the compatibility and the differences.

44.1. Overview of A/D Converter and A/D Range Comparator

The A/D converter converts analog input voltages into digital values and provides the following features. On some devices, any ADC channel can be connected to one of 4 Range Comparators.

44.1.1 Features of the A/D Converter:

Table 44.1-1 gives an overview about the A/D converter features.

Table 44.1-1 A/D converter feature list

Feature	Description	Available on
A/D conversion type	RC type successive approximation conversion with sample & hold circuit	all devices
Conversion time	minimum 1μs per channel	
Resolution	10-bit or 8-bit selectable	
Number of A/D converter macros and analog channels ¹	A/D converter macro (ADC0) with up to 32 analog channels	
	Additional A/D converter macro (ADC1) with up to 22 analog channels	MB91FV460B, MB91F467P, MB91F469Q only
Global A/D channel enable (ADCHE)	Analog channels can be enabled independently of PFR/EPFR setting. This allows conversion while the port is digital output, for example to check for external shortage.	MB91FV460B, MB91F467E, MB91F467P, MB91F469Q only
Result registers	1 common result data register with overwrite protection	all devices
	32 dedicated channel data registers with overwrite protect.	MB91FV460B, MB91F467E, MB91F467P only
A/D Range Comparator (RCO)	4 digital comparators to compare the conversion result with an upper and lower limit	
Conversion start trigger	by software, external trigger pin (GP16[7]=ATGX) or Reload Timer RLT7 ²	all devices
Single conversion mode	Convert the specified channel(s) only once	
Continuous conversion mode	Repeatedly conversion of the specified channels, read out the data after each conversion	
Scan conversion mode	Repeatedly conversion of the specified channels, read out the data after conversion of the last channel (using the dedicated channel data registers)	MB91FV460B, MB91F467E, MB91F467P only
Stop mode	Convert one channel, then temporarily halt until the next activation. ³	all devices

Feature	Description	Available on
Interrupt request to CPU	after end of each single conversion	all devices
	after scan conversion (after the end channel has been converted)	MB91FV460B, MB91F467E, MB91F467P only
DMA access	Interrupts can trigger a DMA access to read out the conversion result register(s)	all devices
Active channel register mode (ACHMD)	ACH[5:0] register shows the number of the channel which is currently under conversion.	all devices
	ACH[5:0] register can additionally show the number of the preceeding channel which was finished before (and caused the currently active interrupt).	MB91FV460B, MB91F467E, MB91F467P, MB91F469Q only
Analog power supply	Separated AV_{CC5} , $AVRH5$, AV_{SS} for ADC0 and ADC1	MB91FV460B only
	Common AV_{CC5} , $AVRH5$, AV_{SS} for ADC0 and ADC1	MB91F467P, MB91F469Q only
	$AVCC5$, $AVRH5$, $AVSS$ for ADC 0	all other devices

1. For the available analog channels on each MB91460 series device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).
2. If 2 ADC macros are mounted, ATGX and RLT7 are connected to both macros.
3. Enables synchronization of the conversion start timing.

44.1.2 Features of the A/D Range Comparator (RCO):

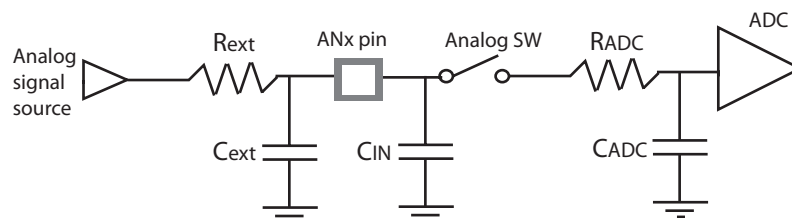
- 4 conversion result Range Comparator channels, comparing the upper 8 bit of the conversion result with an upper and a lower threshold. The thresholds are programmable for the 4 comparators independently.
- Any ADC channel can be connected to one of the 4 range comparators.
- The comparison results will set “overflow” and “interrupt” flags per ADC channel, depending on the configuration. It is possible to configure the comparison for:
 - “out of range”: The flags are set if the A/D result is below the lower OR above the upper threshold,
 - “inside range”: The flags are set if the A/D result is above the lower AND below the upper threshold.
- Range comparison can be followed by an A/D Range Comparator interrupt request to CPU

For details, please refer to section [44.12. Range Comparator Overview \(Page No.1077\)](#).

44.1.3 A/D Converter Input Impedance

The following figure shows the sampling circuit of the A/D converter:

Figure 44.1-1 A/D Converter sampling circuit

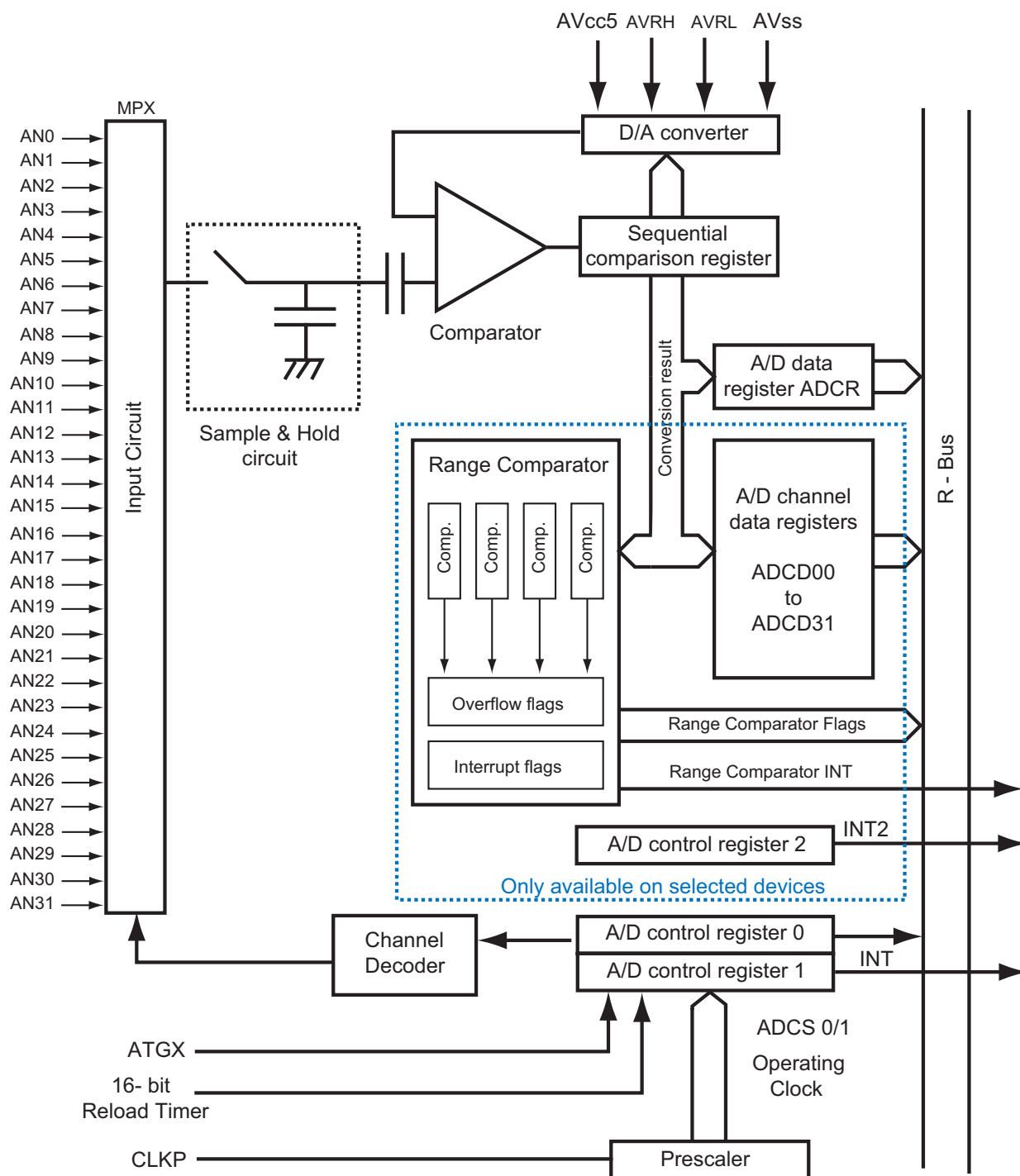


Note: R_{ext} / C_{ext} influence the minimum sampling time. Details see section [44.5.2 Accuracy and Setting of the Sampling time \(Page No.1057\)](#).

44.2. A/D Converter Block Diagram

The following figure shows the block diagram of the A/D converter including the Range Comparator.

Figure 44.2-1 A/D Converter block diagram



44.3. Registers of the A/D Converter and Range Comparator

The A/D converter with Range Comparator has the following registers:

Table 44.3-1 A/D Converter register list

Address (ADC0)	Address (ADC1 *1)	x=0 or 1 for ADC0, ADC1 *1 respectively				Register
		+0	+1	+2	+3	
000498 _H		PORTEN	-	-	-	Global port enable register
0001A0 _H	0005E0 _H	ADxERH (ADERH) *2		ADxERL (ADERL)		A/D channel Enable register
0001A4 _H	0005E4 _H	ADxCS1 (ADCS1)	ADxCS0 (ADCS0)	ADxCR1 (ADCR1)	ADxCR0 (ADCR0)	A/D Control / Status register 0 + 1, A/D Conversion Result register
0001A8 _H	0005E8 _H	ADxCT1 (ADCT1)	ADxCT0 (ADCT0)	ADxSCH (ADSCH)	ADxECH (ADECH)	Sampling timer setting register, Start Channel setting register, End Channel setting register
0006B0 _H	0006DC _H	ADxCS2	-	-	-	A/D Control / Status register 2
000688 _H	0006B4 _H	RCOxH0	RCOxL0	RCOxH1	RCOxL1	Range Comparator 0,1 High/Low threshold registers
00068C _H	0006B8 _H	RCOxH2	RCOxL2	RCOxH3	RCOxL3	Range Comparator 2,3 High/Low threshold registers
000690 _H	0006BC _H	RCOxIRS				Range Comparator Inverted Range Select
000694 _H	0006C0 _H	RCOxOF				Range Comparator Overflow flags
000698 _H	0006C4 _H	RCOxINT				Range Comparator Interrupt flags
0006A0 _H	0006CC _H	ADxCC0	ADxCC1	ADxCC2	ADxCC3	ADC Channel control for ch 0 to 7
0006A4 _H	0006D0 _H	ADxCC4	ADxCC5	ADxCC6	ADxCC7	ADC Channel control for ch 8 to 16
0006A8 _H	0006D4 _H	ADxCC8	ADxCC9	ADxCC10	ADxCC11	ADC Channel control for ch 16 to 23
0006AC _H	0006D8 _H	ADxCC12	ADxCC13	ADxCC14	ADxCC15	ADC Channel control for ch 24 to 31
0006E0 _H	000720 _H	ADCxD0		ADCxD1		ADC Channel Data register, channel 0,1
0006E4 _H	000724 _H	ADCxD2		ADCxD3		ADC Channel Data register, channel 2,3
0006E8 _H	000728 _H	ADCxD4		ADCxD5		ADC Channel Data register, channel 4,5
0006EC _H	00072C _H	ADCxD6		ADCxD7		ADC Channel Data register, channel 6,7
0006F0 _H	000730 _H	ADCxD8		ADCxD9		ADC Channel Data register, channel 8,9
0006F4 _H	000734 _H	ADCxD10		ADCxD11		ADC Channel Data register, channel 10,11
0006F8 _H	000738 _H	ADCxD12		ADCxD13		ADC Channel Data register, channel 12,13
0006FC _H	00073C _H	ADCxD14		ADCxD15		ADC Channel Data register, channel 14,15
000700 _H	000740 _H	ADCxD16		ADCxD17		ADC Channel Data register, channel 16,17
000704 _H	000744 _H	ADCxD18		ADCxD19		ADC Channel Data register, channel 18,19
000708 _H	000748 _H	ADCxD20		ADCxD21		ADC Channel Data register, channel 20,21
00070C _H	00074C _H	ADCxD22		ADCxD23		ADC Channel Data register, channel 22,23
000710 _H	000750 _H	ADCxD24		ADCxD25		ADC Channel Data register, channel 24,25
000714 _H	000754 _H	ADCxD26		ADCxD27		ADC Channel Data register, channel 26,27

Address (ADC0)	Address (ADC1 *1)	x=0 or 1 for ADC0, ADC1 *1 respectively				Register
		+0	+1	+2	+3	
000718 _H	000758 _H	ADCxD28		ADCxD29		ADC Channel Data register, channel 28,29
00071C _H	00075C _H	ADCxD30		ADCxD31		ADC Channel Data register, channel 30,31

1. ADC1 only exists on selected devices, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#).
2. Register names in braces, like “(ADERH)”, show the old ADC0 register names known from the preceding hardware manual revisions.

44.3.1 Global Port Input Enable Register (PORTEN.ADCHE)

The PORTEN register controls the global enabling of all I/O ports after reset (see [30.4.3 Port Input Enable \(Page No.596\)](#)) as well as the enabling of the A/D Converter analog channels.

The global ADC channel enable feature makes the ADC analog inputs independent of PFR/EPFR settings. The feature is available on selected devices only (see “ADCHE” in [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#)) and was introduced for 2 reasons:

- The new ADC1 channels are assigned to ports whose PFR/EPFR combinations are already used completely for other resources, so PFR/EPFR could not be used to enable the analog input.
- Customers may measure digital output signals with the ADC, to check for external shortages, but PFR/EPFR settings for ADC always disable the digital output.

- **PORTEN** : Address 0498_H Access: Byte

7	6	5	4	3	2	1	0	Bit
-	-	-	-	-	ADCHE	CPORTEN	GPORTEN	
X	X	X	X	X	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	RX, W0	RX, W0	R, W	R, W	R, W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\)](#).)

For a full register description, please refer to [30.4.3 Port Input Enable \(Page No.596\)](#).

- **Bit2: ADCHE Global A/D Channel Enable.**

ADCHE	Function
0 [initial]	Global A/D Channel Enable is OFF. The ADC analog lines of channels 0-31 are enabled by setting of the ADC enable bits (ADEn) in the ADERH, ADERL register and PFR/EPFR. PFR/EPFR will set the digital output to HiZ mode and disable the digital input lines of the port.
1	Global A/D Channel Enable is ON. The ADC analog lines of certain channels are enabled by setting of the ADC enable bits (ADEn) in the ADERH, ADERL register only. ADEn will disable the digital input lines of the port, but the digital outputs are not changed. For analog measurement, the user has to switch the port to input direction.

- This bit is cleared by software reset (RST) and can be written and read by CPU.

Note: For the channels of ADC1 (AN32 to AN53, device depending), the ADCHE feature is always ON.

For channels of ADC0 (AN0 to AN31), the ADCHE feature is always OFF if the channels are re-located to other pins (on MB91F467P and MB91FV460B in Series Modes B,P/T).

44.3.2 A/D Input Enable Register (ADER/ADxER)

This register enables the analog input functions of the A/D converter.

- **AD0ERH (ADERH):** Address 01A0_H Access: Word, Half-word, Byte
- **AD1ERH :** Address 05E0_H Access: Word, Half-word, Byte

31	30	29	28	27	26	25	24	Bit
ADE31	ADE30	ADE29	ADE28	ADE27	ADE26	ADE25	ADE24	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

23	22	21	20	19	18	17	16	Bit
ADE23	ADE22	ADE21	ADE20	ADE19	ADE18	ADE17	ADE16	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- **AD0ERL (ADERL) :** Address 01A2_H Access: Word, Half-word, Byte
- **AD1ERL :** Address 05E2_H Access: Word, Half-word, Byte

15	14	13	12	11	10	9	8	Bit
ADE15	ADE14	ADE13	ADE12	ADE11	ADE10	ADE9	ADE8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section 1.5.4 Meaning of Bit Attribute Symbols (Page No.15).)

- **[ADE31-0]: A/D Input Enable**

ADE _n	PORTEN. ADCHE	Function
0 [initial]	X	Analog input of A/D channel n is disabled. The ADC will not sample/convert this channel.
1	0 [initial]	Analog input of the channel n is enabled. Additionally, the port function register (PFR,EPFR) of the corresponding port must be set to enable the analog channel. The PFR/EPFR will switch the port to input direction (output driver = HiZ) and disable the digital input lines.
	1	Analog input of the channel n is enabled. Setting the port function register(s) is not necessary. ADE _n will disable the digital input lines of the ports, but it does not change the port's direction and the port's output signal.

- Software reset (RST) clears ADE_n and PORTEN.ADCHE to '0'.
- Be sure to set start channel and end channel to cover all enabled channels.

MB91460 Series**44.3.3 A/D Control Status Register (ADxCS2)**

The A/D control status registers control and show the status of A/D converter. Do not overwrite ADCS0 register during A/D converting.

- **AD0CS2** : Address 06B0_H Access: Byte
- **AD1CS2** : Address 06DC_H Access: Byte

15	14	13	12	11	10	9	8	Bit
BUSY	INT	INTE	PAUS	-	-	INT2	INTE2	
0	0	0	0	0	0	0	0	Initial value
R(RM1)	R(RM1)	R	R	R0	R0	R(RM1)/W	R/W	Attribute

(For the attributes, refer to section 1.5.4 Meaning of Bit Attribute Symbols (Page No.15).)

- **[bits 15:12] BUSY, INT, INTE, PAUS**

These bits are a mirror of the corresponding bits in ADCS1, intended to quickly read out all status and interrupt information using only one register access. To write the bits, access them via ADCS1.

- Read-modify-write instructions read the bits BUSY and PAUS as “1”.

- **[bits 11:10] -**

These bits do not exist. Read operation returns “0”.

- **[bit 9] INT2 (End of Scan Flag)**

The End of Scan flag is set in continuous conversion mode, when conversion data of the last channel is stored in ADCR, whereas the last channel is defined by ADECH register setting.

- If bit 8 (INTE2) is ‘1’ when this bit is set, an End of Scan interrupt request to CPU is generated, or, if activation of DMA is enabled, DMA is activated.
- Only clear INT2 by writing ‘0’ when A/D conversion is halted.
- Initialized to ‘0’ by a software reset (RST).
- If DMA is used, this bit is cleared at the end of DMA transfer.
- Read-modify-write operations read this bit as ‘1’.

- **[bit 8] INTE2 (Enable End of Scan Interrupt)**

INTE2 enables the End of Scan interrupt request to CPU and/or DMA.

INTE2	Function
0 [initial]	Disable End of Scan interrupt request.
1	Enable End of Scan interrupt reques.

- Initialized to ‘0’ by a software reset (RST).

44.3.4 A/D Control Status Register (ADCS1/ADxCS1)

- **AD0CS1 (ADCS1)** : Address 01A4_H Access: Half-word, Byte
- **AD1CS1** : Address 05E4_H Access: Half-word, Byte

15	14	13	12	11	10	9	8	Bit
BUSY	INT	INTE	PAUS	STS1	STS0	STRT	reserved	
0	0	0	0	0	0	0	0	Initial value
R(RM1)/W	R(RM1)/W	R/W	R/W	R/W	R/W	R(RM0)/W	R/W	Attribute

(For the attributes, refer to section 1.5.4 Meaning of Bit Attribute Symbols (Page No.15).)

- **[bit 15] BUSY (busy flag and stop)**

BUSY	Function
Reading	A/D converter operation indication bit. Set on activation of A/D conversion and cleared on completion.
Writing	Writing '0' to this bit during A/D conversion forcibly terminates conversion. Use to forcibly terminate in continuous and stop modes.

- Read-modify-write instructions read the bit as '1'.
- Cleared on the completion of A/D conversion in single conversion mode.
- In continuous and stop mode, the flag is not cleared until conversion is terminated by writing '0'.
- Initialized to '0' by a software reset (RST).
- Do not specify forcible termination and software activation (BUSY= '0' and STRT= '1') at the same time.

- **[bit 14] INT (End of Conversion Interrupt flag)**

This bit is set when conversion data is stored in ADCR.

- If bit 5 (INTE) is '1' when this bit is set, an interrupt request to CPU is generated, or, if activation of DMA is enabled, DMA is activated.
- Only clear this bit by writing '0' when A/D conversion is halted.
- Read-modify-write instructions read the bit as '1'.
- Initialized to '0' by a software reset (RST).
- If DMA is used, this bit is cleared at the end of DMA transfer.

- **[bit 13] INTE (End of Conversion Interrupt enable)**

This bit enables or disables the conversion completion interrupt.

INTE	Function
0	Disable interrupt request [Initial value]
1	Enable interrupt request

- Cleared by a software reset (RST).

- [bit 12] **PAUS (A/D converter pause)**

This bit is set when A/D conversion temporarily halts.

The A/D converter has one register to store the conversion result (ADxCR) and optional 32 ADC channel data registers. If a conversion is finished and the data of the **preceding** conversion has not been read out before, preceding data would be overwritten.

To avoid this problem, the next conversion data is not stored in the data registers until the preceding value has been read out (e.g. by DMA). A/D conversion halts during this time and the data registers are protected.

To resume the conversion, the interrupt flags INT, INT2 must be cleared after reading out the data.

Table 44.3-2 A/D converter pause and resume

Mode	INTE2	INTE	PAUS is set when...
any	0	0	PAUS is not set because no interrupt request is generated (although the flags may be set, but interrupt to CPU/DMA is not enabled).
Continuous	0	1	PAUS is set after each channel conversion, when the interrupt request INT of the preceding conversion has not been cleared. To resume conversion, the INT flag must be cleared.
	1	0	PAUS is set after the conversion of the start channel (defined by ADxSCH register), when the interrupt request INT2 of the preceding scan has not been cleared. To resume conversion, INT2 flag must be cleared.
	1	1	PAUS is set after each channel conversion when the interrupt request INT of the preceding conversion or the interrupt request INT2 of the preceding scan have not been cleared. To resume conversion, INT and INT2 flags must be cleared.
other	X	1	PAUS is set after each channel conversion when the interrupt request INT of the preceding conversion has not been cleared. To resume conversion, INT flag must be cleared.

- PAUS is only a flag indicating that the conversion is temporary halted. It does not control the conversion state machine. To resume the conversion, clear the INT flags together with PAUS.
- PAUS is cleared by writing '0' or by a reset.
If '0' is written and the setting condition in [Table 44.3-2](#) is still valid, PAUS is set again.
- PAUS is not cleared after DMA access. DMA access clears the interrupt flags which allows to resume the conversion.

- [bit 11, 10] **STS1, STS0 (Start source select)**

These bits select the A/D activation source.

STS1	STS0	Function
0	0	Software activation [Initial value]
0	1	External trigger pin activation and software activation
1	0	Timer activation and software activation
1	1	External trigger pin activation, timer activation and software activation

- These bits are initialized '00' by software reset (RST).
- In multiple-activation modes, the first activation to occur starts A/D conversion.
- The activation source changes immediately on writing to the register. Therefore care is required when

switching activation mode during A/D operation.

- The A/D converter detects falling edges on the external trigger pin. When external trigger level is 'L' and if these bits are changed to external trigger activation mode, A/D conversion may start.
- Selecting the timer selects the 16-bit Reload Timer 7. If 2 ADC macros are mounted, RLT7 is used for both ADCs.

- **[bit 9] STRT (Start)**

Writing '1' to this bit starts A/D conversion (software activation).

- Write '1' again to restart conversion.
- After writing '1', the bit is cleared by hardware after the next CLKP cycle.
- Read-modify-write operation reads '0'.
- Initialized to '0' by a software reset (RST).
- In continuous and stop mode, restarting is not occurred. Check BUSY bit before writing '1'. (Activate conversion after clearing.)
- Do not specify forcible termination and software activation (BUSY='0' and STRT='1') at the same time.

- **[bit 8] reserved bit**

Always write '0' to this bit.

MB91460 Series**44.3.5 A/D Control Status Register (ADCS0/ADxCS0)**

- **AD0CS0 (ADCS0)** Address 01A5_H Access: Half-word, Byte. Read-modify-write access is not allowed
- **AD1CS0 :** Address 05E5_H Access: Half-word, Byte. Read-modify-write access is not allowed

7	6	5	4	3	2	1	0	Bit
MD1	MD0	S10	ACH4	ACH3	ACH2	ACH1	ACH0, ACHMD *1	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R	R	R	R	R,W	Attribute

1. ACHMD bit is available on selected devices only, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#).

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\)](#).)

- **[bit 7, 6] MD1, MD0 (A/D converter mode set)**

These bits the operation mode.

MD1	MD0	Operating mode
0	0	Single mode 1 (Reactivation during A/D conversion is allowed)
0	1	Single mode 2 (Reactivation during A/D conversion is not allowed)
1	0	Continuous mode or Scan conversion mode (Reactivation during A/D conversion is not allowed)
1	1	Stop mode (Reactivation during A/D conversion is not allowed)

- **Single mode:** A/D conversion is continuously performed from the selected start channel (ADSCH) to the selected end channel (ADECH). The conversion stops once it has been done for all these channels.
- **Continuous mode:** A/D conversion is repeatedly performed from the selected start channel (ADSCH) to the selected end channel (ADECH) in a row.
- **Stop mode:** A/D conversion is performed from the selected start channel (ADSCH) to the selected end channel (ADECH), followed by a pause after each channel. The conversion is resumed upon activation.

For details about the conversion modes, please refer to section [44.6. Operation Modes of A/D Converter \(Page No.1058\)](#)

When A/D conversion is started in continuous mode or stop mode, conversion operation continues until stopped by the BUSY bit. Conversion is stopped by writing '0' to the BUSY bit.

On activation after forcibly stopping, conversion starts from the start channel, selected by ADSCH register.

Reactivation during A/D conversion is disabled for any of the timer, external trigger and software start sources in single mode 2, continuous and stop mode.

- **[bit 5] S10**

This bit defines resolution of A/D conversion. If this bit set '0', the resolution is 10-bit. In the other case, resolution is 8-bit.

- Initialized to '0' by a reset.

Note: Please see the details in section [44.10.2 How to specify a bit length \(resolution\) \(Page No.1070\)](#).

- **[bit 4 to 0] ACH4-0 Analog convert select channel (active channel), read-only**

These bits show the number of the currently or previously converted analog channel, depending on bit ACHMD (see below).

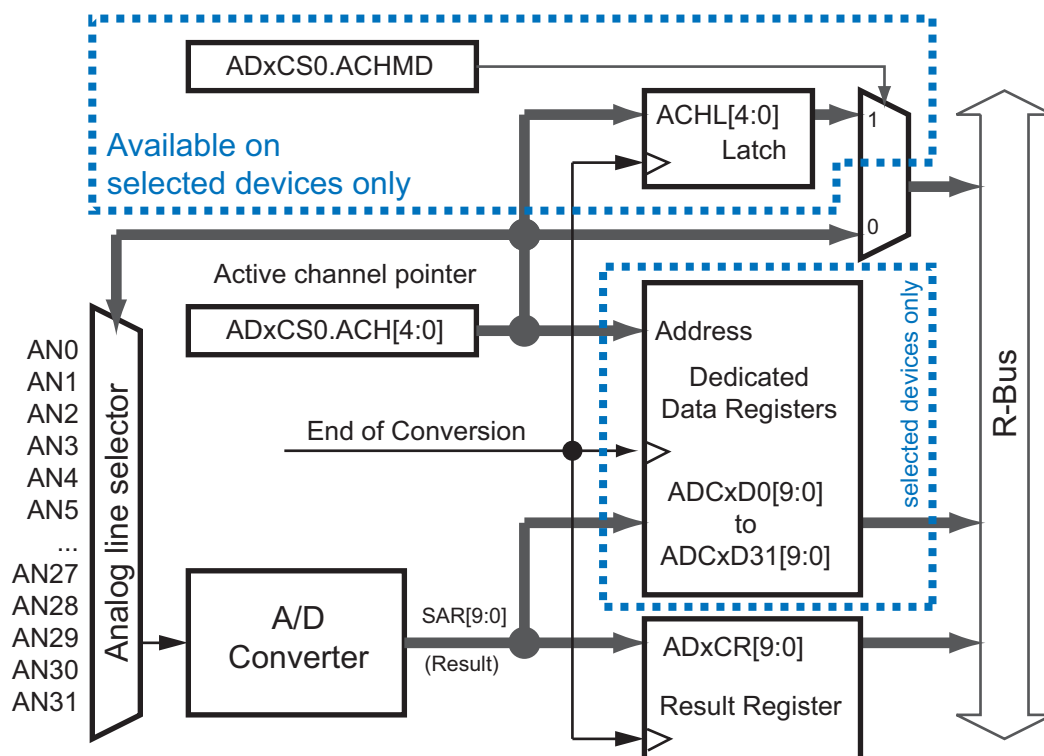
ACH4	ACH3	ACH2	ACH1	ACH0	Converted channel
0	0	0	0	0	AN0
0	0	0	0	1	AN1
...					...
1	1	1	1	0	AN30
1	1	1	1	1	AN31

- Writing these bits has no effect (bit 0 is writeable with special function ADCHMD).
- Initialized to "0000" by software reset (RST).

- **[bit 0] ACHMD (ACH register mode, write-only)**

The ACHMD bit is available on selected devices only, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#). On those selected devices, there is a latch which stores the active channel information (ACH[4:0]) when the conversion result data is stored, see [Figure 44.3-1](#). The CPU can read out ACH[4:0] directly, or it can read the latch.

Figure 44.3-1 A/D Converter ACH[5:0] Latch



- ACHMD is a write-only bit.
- Read- or read-modify-write access returns the value of bit ACH0, that's why read-modify-write access is not

MB91460 Series

allowed.

- Initial value is 0.

The behaviour depends on the conversion mode and the control bit ACHMD. If the ACHMD feature is not implemented, the CPU always reads non-latched ACH[4:0] like with ADCMD=0.

Table 44.3-3 A/D Converter ACHMD functionality

Conversion mode	State at the time when ACH[4:0] is read	ACHMD	ACH[4:0] reads the number of the ADC channel...
all	during a conversion is ongoing or during pause (PAUS flag = 1)	0 [initial]	which is <i>currently</i> under conversion
		1	which was <i>finished previously</i>
all	after a conversion has been forcibly stopped (set BUSY=0)	0 [initial]	which <i>just</i> has been <i>cancelled</i>
		1	which was <i>finished previously</i>
Stop mode	after a conversion is done (INT flag = 1)	0 [initial]	which <i>just</i> has been <i>converted</i>
		1	
Single shot mode	after end channel conversion is done (INT=1 and BUSY=0)	0 [initial]	which <i>just</i> has been <i>converted</i>
		1	
Single shot / Continuous mode	after a conversion is done and the next channel conversion is not started yet ¹	0 [initial]	which <i>just</i> has been <i>converted</i>
		1	

1. If not all channels between start and end channel are enabled, the state machine needs some clock cycles to count up to the next enabled channel.

Example: AN3 is enabled, AN4-19 are disabled, AN20 is enabled

--> the state machine needs 17 CLKP cycles to count up to AN20.

44.3.6 Common Data Register (ADCR1/ADxCR1, ADCR0/ADxCR0)

These registers store the conversion results of the A/D converter. ADCR0 stores lower 8-bit. ADCR1 stores upper 2-bit. The register values are updated at the completion of each conversion. The registers normally store the results of the previous conversion.

• **AD0CR1 (ADCR1)** : Address 01A6_H Access: Word, Half-word, Byte

• **AD1CR1** : Address 05E6_H Access: Word, Half-word, Byte

15	14	13	12	11	10	9	8	Bit
-	-	-	-	-	-	D9	D8	
0	0	0	0	0	0	X	X	Initial value
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R	R	Attribute

• **AD0CR0 (ADCR0)** : Address 01A7_H Access: Word, Half-word, Byte

• **AD1CR0** : Address 05E7_H Access: Word, Half-word, Byte

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R	R	R	R	R	R	R	R	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\)](#).)

- Bit 15 to 10 of ADCR1 are read as '0'.
- The A/D converter has a conversion data protection function. See section [44.8. Protection of the ADC Results \(Page No.1066\)](#) and the description of the PAUS bit in section [44.3.4 A/D Control Status Register \(ADCS1/ADxCS1\) \(Page No.1044\)](#).

44.3.7 Dedicated A/D Channel Data Register (ADCxD0 to ADCxD31)

There are 32 ADC result data registers, one per channel. The registers are written by hardware at the end of conversion of the attached channel. ADCD0 is attached to channel 0, ADCD31 is attached to channel 31.

- **ADC0D0 ... ADC0D31** : Address 06E0_H to 071E_H Access: Word, Half-word, Byte
- **ADC1D0 ... ADC1D31** : Address 0720_H to 075E_H Access: Word, Half-word, Byte

15	14	13	12	11	10	9	8	Bit
-	-	-	-	-	-	D9	D8	
0	0	0	0	0	0	X	X	Initial value
R0	R0	R0	R0	R0	R0	R	R	Attribute
7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R	R	R	R	R	R	R	R	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\).](#))

- Bit 15 to 10 of the ADCD registers are read as '0'.
- The A/D converter has a conversion data protection function. In continuous conversion mode, the protection function can be changed to protect the A/D Channel Data registers rather than the A/D Data Register (ADxCR1). See section [“44.8. Protection of the ADC Results”](#) for further information.

44.3.8 Sampling Timer Setting Register (ADCT/ADxCT1)

ADCT register controls the sampling time and comparison time of analog input. This register sets A/D conversion time. Do not update value of this register during A/D conversion operation.

• **AD0CT1 (ADCT1):** Address 01A8_H Access: Word, Half-word, Byte

• **AD1CT1:** Address 05E8_H Access: Word, Half-word, Byte

15	14	13	12	11	10	9	8	Bit
CT5	CT4	CT3	CT2	CT1	CT0	ST9	ST8	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

• **AD0CT0 (ADCT0):** Address 01A9_H Access: Word, Half-word, Byte

• **AD1CT0:** Address 05E9_H Access: Word, Half-word, Byte

7	6	5	4	3	2	1	0	Bit
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0	
0	0	1	0	1	1	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\)](#).)

• **[bit 15 to 10] CT5-0 (A/D comparison time set)**

These bits specify clock division of comparison time.

- Setting '000001' means one division (=CLKP).
- Do not set these bits '000000'.
- Initialized these bits to '000100' by software reset (RST).
- Comparison time = CT value * CLKP cycle * 10 + (4 * CLKP)
- For details please see section [44.5.3 Setting the Comparison Time \(Page No.1058\)](#).

• **[bit 9 to 0] ST9-0 (Analog input sampling time set)**

These bits specify sampling time of analog input.

- Initialized these bits to '0000101100' (44) by software reset (RST).
- Sampling time = ST value * CLKP cycle
- For details please see section [44.5.2 Accuracy and Setting of the Sampling time \(Page No.1057\)](#).

44.3.9 A/D Channel Setting Register (ADSCH/ADxSCH, ADECH/ADxECH)

These registers specify the channels for the A/D converter to convert. Do not update these registers while the A/D converting is operating.

- **AD0SCH (ADSCH)** : Address 01AA_H Access: Word, Half-word, Byte
- **AD1SCH** : Address 05EA_H Access: Word, Half-word, Byte

15	14	13	12	11	10	9	8	Bit
-	-	-	ANS4	ANS3	ANS2	ANS1	ANS0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

- **AD0ECH (ADECH)** : Address 01AB_H Access: Word, Half-word, Byte
- **AD1ECH** : Address 05EB_H Access: Word, Half-word, Byte

7	6	5	4	3	2	1	0	Bit
-	-	-	ANE4	ANE3	ANE2	ANE1	ANE0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section 1.5.4 Meaning of Bit Attribute Symbols (Page No.15).)

- **[bit 12 to 8] ANS[4:0] Start channel**
- **[bit 4 to 0] ANE[4:0] End channel**

These bits specify the start channel and the the end channel for A/D conversion.

Always make sure that the start channel and the end channel are enabled in ADxERH/ADxERL register!

- In all conversion modes, conversion is performed from the start channel (ANS[4:0]) up to the end channel (ANE[4:0]). Conversion then starts again from the start channel.
- Only those channels are converted which are enabled (ADE[n] is set in ADxERH/ADxERL register). Non-enabled channels are skipped. Example:
Channel Setting ANS=0, ANE=7, AD0ERL=0x00C3, single conversion mode
Operation: Conversion channel AN0 -> AN1 -> AN6 -> AN7 end
- Setting ANE[4:0] the same channel as in ANS[4:0] specifies conversion for that single channel only.
- If ANS[4:0] > ANE[4:0], conversion starts with the channel specified by ANS, continues up to channel 31, starts again from channel 0, and ends with the channel specified by ANE. Example:
Channel Setting ANS=30, ANE=3, single conversion mode:
Operation : Conversion channel AN30 -> AN31 -> AN0 -> AN1 -> AN2 -> AN3 end
- Initialized to ANS="00000", ANE="00000" by a software reset (RST).

ANS4 ANE4	ANS3 ANE3	ANS2 ANE2	ANS1 ANE1	ANS0 ANE0	Start / End Channel
0	0	0	0	0	AN0
0	0	0	0	1	AN1
0	0	0	1	0	AN2
...					...
1	1	1	1	0	AN30
1	1	1	1	1	AN31

44.4. Analog Input Connections

This section explains how the analog input lines are enabled. The following registers influence the analog channels:

- Global Port Enable register (PORTEN)
- Port Function Register and Extended Port Function Register (PFR, EPFR)
- ADC Channel Enable registers (ADERH, ADERL)

44.4.1 Global ADC Analog Channel Enable

The global ADC channel enable feature makes the ADC analog inputs independent of PFR/EPFR settings. For details, please refer to section [44.3.1 Global Port Input Enable Register \(PORTEN.ADCHE\) \(Page No.1041\)](#).

The feature is only available on selected devices, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#).

44.4.2 ADC 0 Analog Inputs AN0 to AN31

ADC 0 serves the analog inputs AN0 to AN31. There are 2 methods for enabling the analog inputs:

- Set ADC channel enable bits (ADEn) in the AD0ERH, AD0ERL register and set PFR/EPFR of the attached I/O port.
- Set ADC channel enable bits (ADEn) in the AD0ERH, AD0ERL register and set global ADC channel enable, see [44.3.1 Global Port Input Enable Register \(PORTEN.ADCHE\) \(Page No.1041\)](#).

44.4.3 ADC 1 Analog Inputs AN32 to AN53

ADC 1 serves the analog inputs AN37 to AN53 and is available on selected devices only, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#).

The analog inputs are enabled just by setting the ADC channel enable bits ADEn in the AD1ERH, AD1ERL registers. The Global ADC Analog Channel Enable feature is fixed ON here.

44.5. Accuracy and Parameter Setting of the A/D Converter

This section explains the A/D Converter parameters and the timing settings.

44.5.1 Definitions of A/D Converter Terms

- **Resolution**

Analog variation that is recognizable by the A/D converter.

- **Nonlinearity error**

Deviation between actual conversion characteristics and a straight line connecting the zero transition point ($00\ 0000\ 0000_B \leftrightarrow 00\ 0000\ 0001_B$) and the full scale transition point ($11\ 1111\ 1110_B \leftrightarrow 11\ 1111\ 1111_B$)

- **Differential nonlinearity error**

Deviation of the input voltage from the ideal value that is required to change the output code by 1 LSB.

- **Total error**

This error indicates the difference between actual and theoretical values, including the zero transition error, full scale transition error, and nonlinearity error.

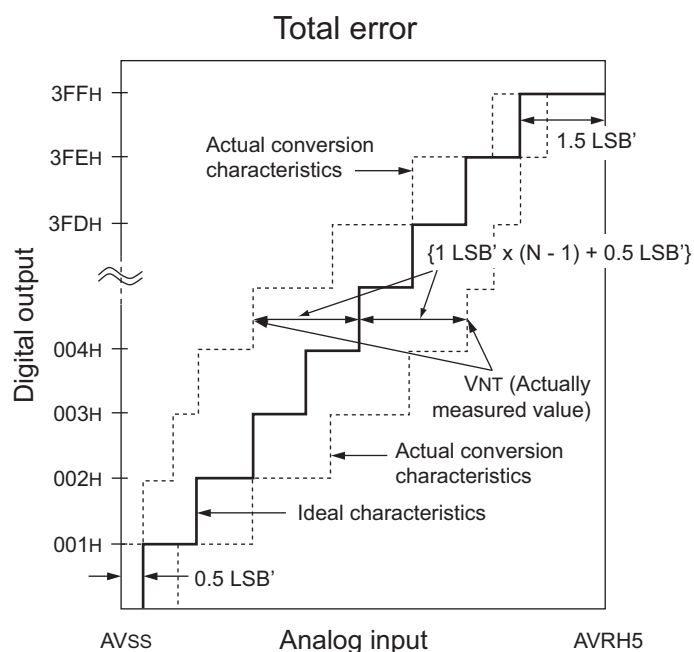
- **Zero reading voltage**

Input voltage which results in the minimum conversion value.

- **Full scale reading voltage**

Input voltage which results in the maximum conversion value.

Figure 44.5-1 Total error of A/D converter



$$1\text{LSB}' \text{ (Ideal value)} = \frac{AVRH - AV_{SS}}{1024} \quad [\text{V}]$$

$$V_{OT}' \text{ (Ideal value)} = AV_{SS} + 0.5\text{LSB} \quad [\text{V}]$$

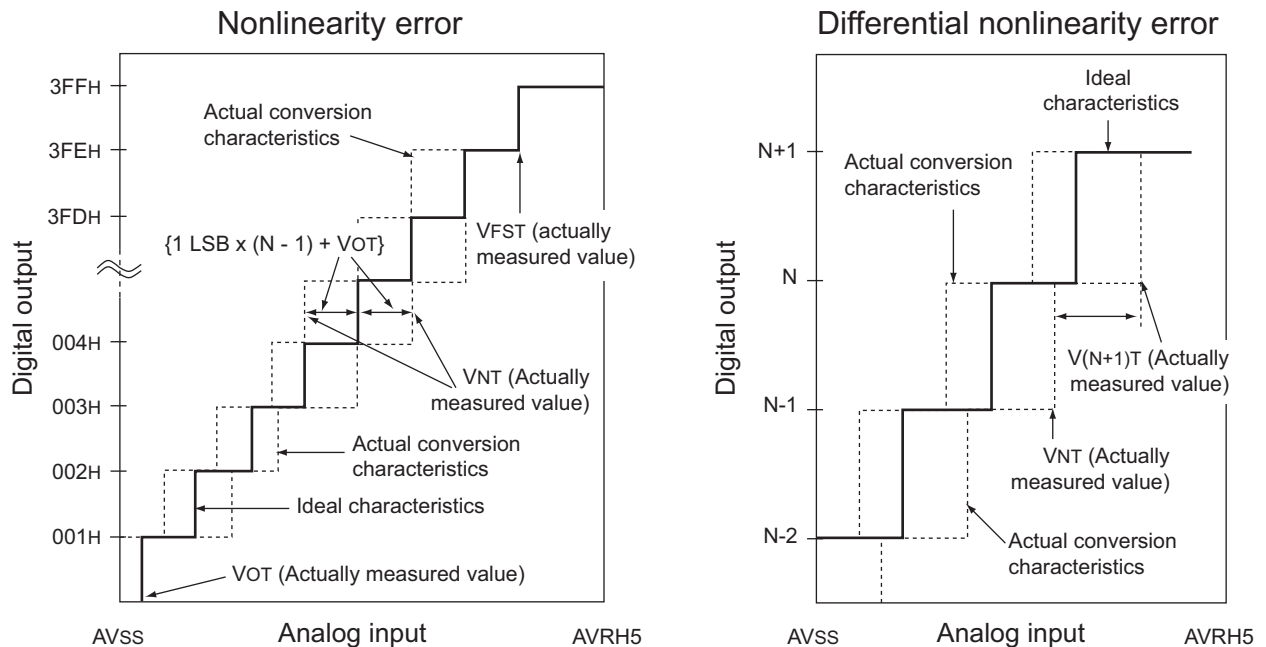
$$V_{FST}' \text{ (Ideal value)} = AVRH5 - 1.5\text{LSB} \quad [\text{V}]$$

$$\text{Total error of digital output } N = \frac{V_{NT} - \{1\text{LSB}' * (N - 1) + 0.5\text{LSB}'\}}{1\text{LSB}'} \quad [\text{LSB}]$$

N: A/D converter digital output value

V_{NT} : Voltage at which the digital output changes from (N+1) to N

Figure 44.5-2 ADC Nonlinearity and Differential Nonlinearity Error



$$1 \text{ LSB} = \frac{V_{FST} - V_{OT}}{1022} \quad [\text{V}]$$

$$\text{Nonlinearity error of digital output } N = \frac{V_{NT} - \{1\text{LSB} * (N - 1) + V_{OT}\}}{1\text{LSB}} \quad [\text{LSB}]$$

$$\text{Differential nonlinearity error of digital output } N = \frac{V_{(N+1)T} - V_{NT}}{1\text{LSB}} - 1 \quad [\text{LSB}]$$

N: A/D converter digital output value

V_{NT} : Voltage at which digital output transit from (N+1) to N

V_{OT} : Voltage at which the digital output changes from 000_H to 001_H

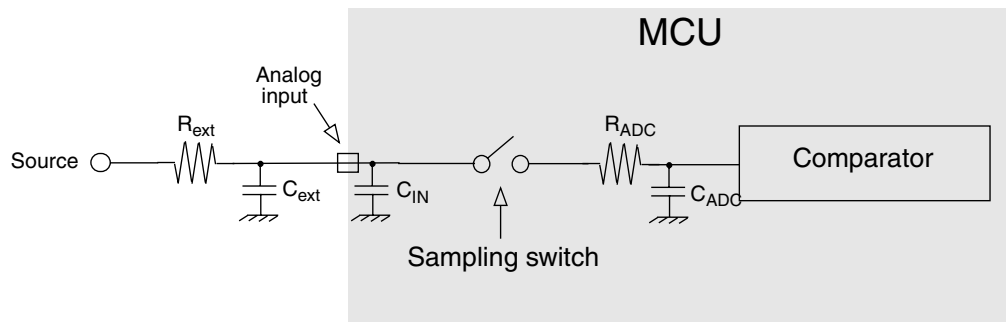
V_{FST} : Voltage at which the digital output changes from 3FE_H to 3FF_H.

44.5.2 Accuracy and Setting of the Sampling time

If the external impedance is too high or the sampling time too short, the analog voltage charged to the internal sample and hold capacitor is insufficient, adversely affecting the A/D conversion precision.

To satisfy the A/D conversion precision, a sufficient sampling time must be selected. The required sampling time depends on the external driving impedance R_{ext} , the board capacitance of the A/D converter input pin C_{ext} and the AVcc5 voltage level. The following replacement model can be used for the calculation:

Figure 44.5-3 A/D Converter sampling replacement model



R_{ext} : external driving impedance
 C_{ext} : capacitance of PCB at A/D converter input
 C_{IN} : capacitance of MCU input pin
 R_{ADC} : resistance within MCU
 C_{ADC} : sampling capacitance within MCU

About the C_{IN} , R_{ADC} and C_{ADC} paramaters, please refer to the MCU datasheets.

The sampling time should be set to minimum “ 7τ ”. The following approximation formula for the replacement model above can be used:

$$\bullet T_{\text{samp}} [\text{min}] = 7 * (R_{\text{ext}} * (C_{\text{ext}} + C_{\text{IN}}) + (R_{\text{ext}} + R_{\text{ADC}}) * C_{\text{ADC}})$$

A certain minimum sampling time $T_{\text{samp}} [\text{min}]$ is required. Please refer to the MCU datasheets.

To set the sampling time, write the bits ADxCT.ST[9:0]. The sampling time will be

$$\bullet T_{\text{samp}} = \text{ST}[9:0] \text{ value} * \text{CLKP cycle.}$$

whereas T_{samp} must be greather then $T_{\text{samp}}[\text{min}]$.

If the sampling time cannot be sufficient, connect a capacitor of about 0.1 μF to the analog input pin. In this case the internal sampling capacitance C_{ADC} will be charged out of this external capacitance.

A big external driving impedance also adversely affects the A/D conversion precision due to the pin input leakage current I_{IL} (static current before the sampling switch) or the analog input leakage current I_{AIN} (total leakage current of pin input and comparator during sampling). The effect of the pin input leakage current I_{IL} cannot be compensated by an external capacitor.

The accuracy gets worse as $|\text{AVRH} - \text{AVRL}|$ becomes smaller.

44.5.3 Setting the Comparison Time

After sampling the analog value, the A/D Converter needs approximately 11 clock cycles for comparison. The register bits ADxCT.CT[5:0] are used to define the prescaler for the A/D Converter clock CKIN, whereas CKIN has the following minimum cycle time:

- $T_{CKIN} [\text{min}] = 60 \text{ ns}$ for $4.5\text{V} \leq A_{VCC} \leq 5.5\text{V}$
- $T_{CKIN} [\text{min}] = 200 \text{ ns}$ for $3.0\text{V} \leq A_{VCC} < 4.5\text{V}$

To set the clock time, write the bits ADxCT.CT[5:0]. The clock time will be:

- $T_{CKIN} = \text{CT}[5:0] \text{ value} * \text{CLKP cycle}$.

whereas T_{CKIN} must be greater than $T_{CKIN}[\text{min}]$.

The comparison time will then be:

- $T_{\text{comp}} = (\text{CT}[5:0] \text{ value} * \text{CLKP cycle} * 10) + (\text{CLKP cycle} * 4)$

For the maximum allowed comparison time, please refer to the MCU datasheets.

44.6. Operation Modes of A/D Converter

The A/D converter operates using the successive approximation method with 10-bit or 8-bit resolution. There is one 16-bit register provided to store conversion results (ADxCR), which is updated each time conversion is completed.

On selected devices (see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#)), there are ADC Channel Data register per channel (ADCxD0...ADCxD31), which are updated each time the assigned channel is converted. The Channel Data registers especially improve the continuous conversion mode.

It is recommended to use the DMA service. The following describes the operation modes.

44.6.1 Operation Mode Overview

A short overview about the operation modes is given in [Table 44.6-1](#):

Table 44.6-1 A/D Converter operation modes

Mode	Description	Available on
Single conversion mode	Convert the specified channel(s) only once	all devices
Continuous conversion mode	Repeatedly conversion of the specified channels, read out the data after each conversion	
Scan conversion mode	Repeatedly conversion of the specified channels, read out the data after conversion of the last channel (using the dedicated channel data registers)	MB91FV460B, MB91F467E, MB91F467P only
Stop mode	Convert one channel, then temporarily halt until the next activation.	all devices

MB91460 Series

44.6.2 Single-Shot Conversion

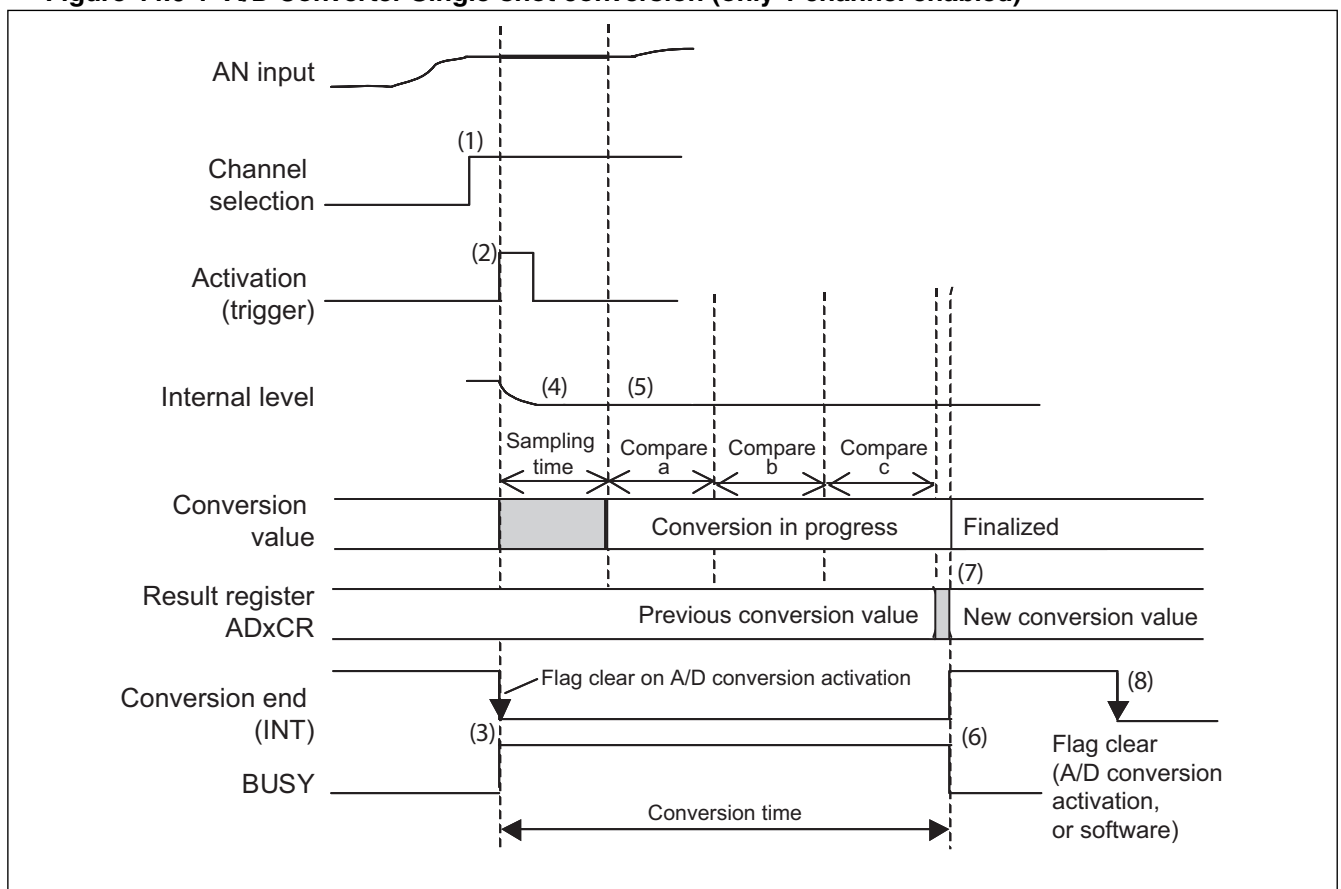
In single conversion mode, the analog input signals selected by the ANS bits and ANE bits are converted in order until the completion of conversion on the end channel determined by the ANE bits. A/D conversion then ends. If the start channel and end channel are the same (ANS=ANE), only a single channel conversion is performed. Channels, which are not enabled in ADxERH/ADxERL register, are skipped.

Examples:

- ANS=00000b, ANE=00011b
Start -> AN0 -> AN1 -> AN2 -> AN3 -> End
- ANS=00010b, ANE=00010b
Start -> AN2 -> End

Figure 44.6-1 shows the operation of A/D converter in Single-shot conversion mode, if only 1 channel is converted (start channel = end channel):

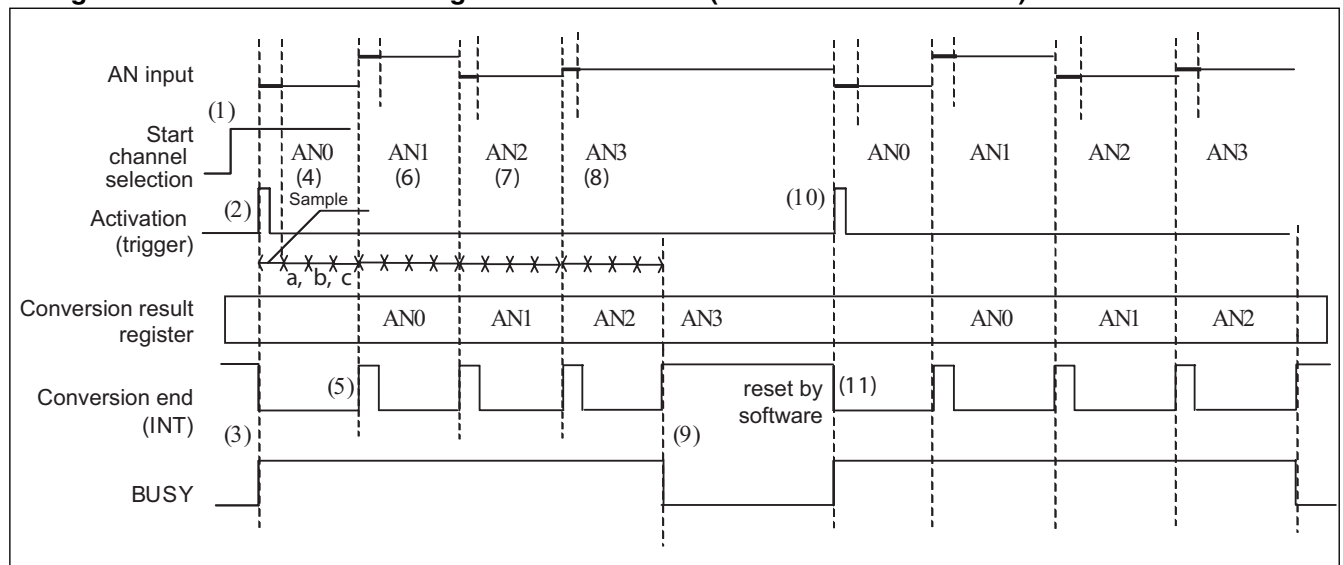
Figure 44.6-1 A/D Converter Single-shot conversion (only 1 channel enabled)



- (1) Channel selection
- (2) A/D conversion activation (Trigger input: Software trigger/Reload timer/External trigger)
- (3) INT flag clear, BUSY flag set
- (4) Sampling time
- (5) Comparison (Comparison a + Comparison b + Comparison c)
- (6) Conversion end, INT flag set, BUSY flag clear by hardware (when the end channel was converted)
- (7) Result register is loaded
- (8) INT flag clear by software or after DMA access

Figure 44.6-2 shows the operation of A/D converter in Single-shot conversion mode for more than one channels. In this example, channels AN0 to AN3 are enabled.

Figure 44.6-2 A/D converter Single-shot conversion (channels 0 to 3 enabled)



- (1) Activation channel selection
- (2) A/D activation (Trigger: Software trigger/Reload timer/External trigger)
- (3) INT flag clear, BUSY flag set
- (4) AN0 conversion
 - a. Sampling time, conversion (comparison a + comparison b + comparison c)
 - b. Conversion end
 - c. Write the conversion value to result register.
- (5) INT flag set. INT is then cleared after reading out the conversion result register
- (6) AN1 conversion
- (7) AN2 conversion
- (8) AN3 conversion
- (9) INT flag set, BUSY flag clear by hardware after end channel
- (10) Next A/D activation
- (11) INT flag is cleared by software, BUSY flag set

MB91460 Series

44.6.3 Continuous Conversion Mode

In continuous mode the analog input signals selected by the ANS bits and ANE bits are converted in order until the completion of conversion on the end channel determined by the ANE bits, then the converter returns to the ANS channel for analog input and repeats the process continuously. When the start and end channels are the same (ANS=ANE), conversion is performed continuously for that channel.

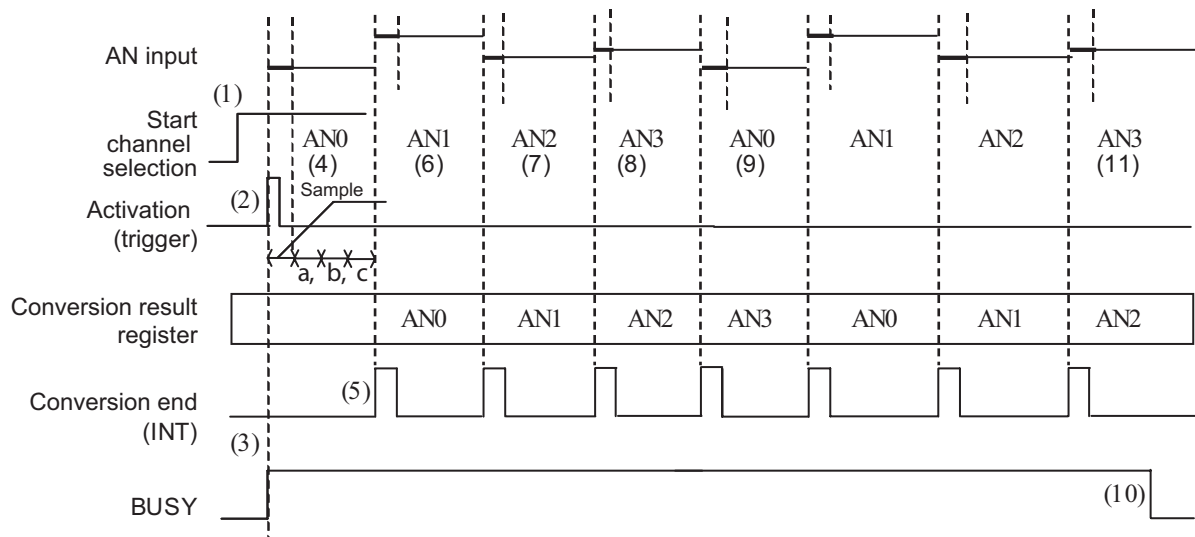
Examples:

- ANS=00000b, ANE=00011b
Start -> AN0 -> AN1 -> AN2 -> AN3 -> AN0 ... -> repeat
- ANS=00010b, ANE=00010b
Start -> AN2 -> AN2 -> AN2 ... -> repeat

In continuous mode, conversion is repeated until '0' is written to the BUSY bit. (Writing '0' to the BUSY bit forcibly stops the conversion operation.) Note that forcibly terminating operation halts the current conversion during mid-conversion. (If operation is forcibly terminated, the value in the conversion register is the result of the most recently completed conversion.)

Figure 44.6-3 shows the A/D converter operation in continous conversion mode. In this example, channels AN0 to AN3 are enabled.

Figure 44.6-3 A/D converter continous conversion (channels 0 to 3 enabled)



- (1) Activation channel selection
- (2) A/D activation (Trigger: Software trigger/Reload timer/External trigger)
- (3) BUSY flag set (the INT flag can be cleared also here)
- (4) AN0 conversion
 - a. Sampling time, conversion (comparision a + comparision b + comparision c)
 - b. Conversion end
 - c. Write the conversion value to result register.
- (5) INT flag set. INT is then cleared after reading out the conversion result register
- (6) AN1 conversion
- (7) AN2 conversion
- (8) AN3 conversion
- (9) Next loop starts with AN0 conversion
- (10) BUSY is forcibly cleared, conversion stopps
- (11) AN3 conversion is cancelled

44.6.4 Scan Conversion Mode

Scan Conversion is available in Continuous Mode on selected devices only (see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#)). Scan conversion converts all channels and stores the results to the dedicated Channel Data registers. After conversion of the end channel, an interrupt request INT2 to CPU or DMA can be generated. Therefore, scan conversion is a kind of Continuous Mode with different result data handling.

Table 44.6-2 A/D Continuous mode versus Scan mode

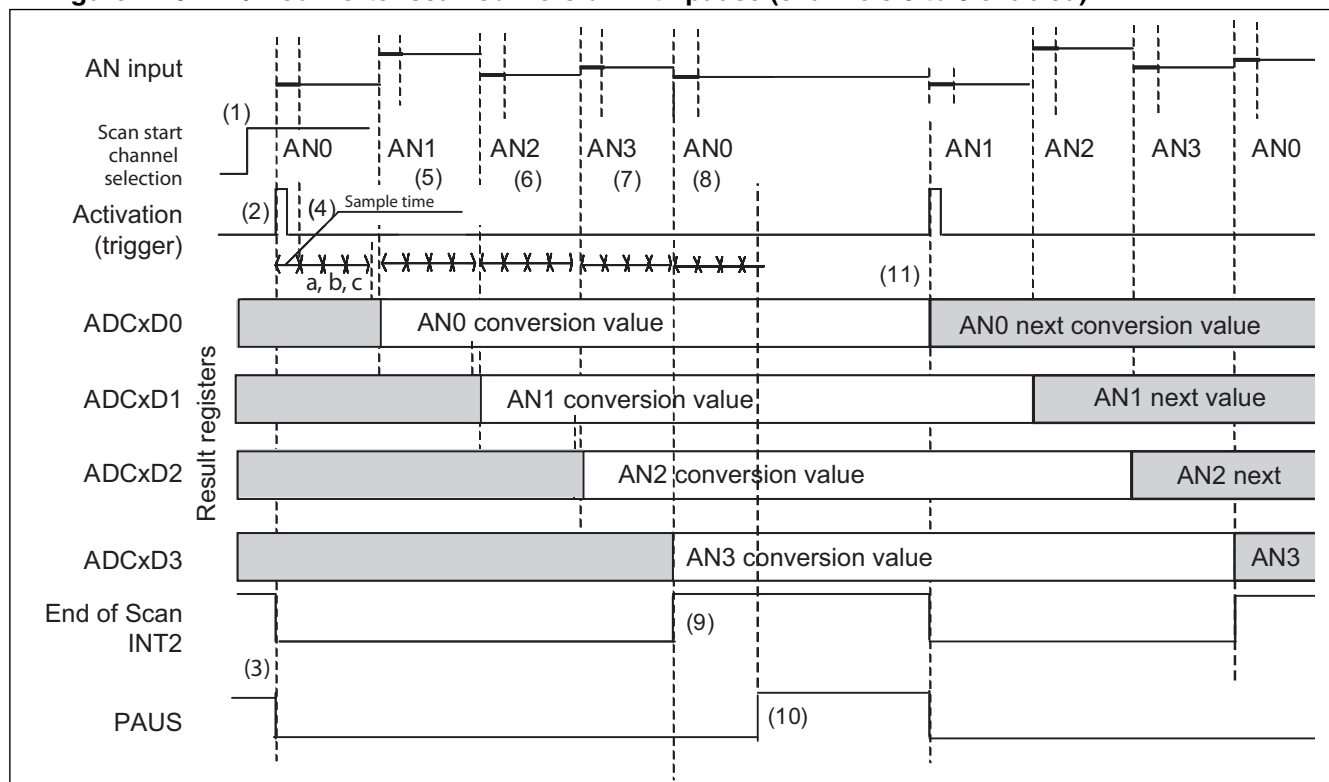
Mode	Description
Continuous conversion	Repeatedly conversion of the specified channels, use “End of Conversion” interrupt INT, read out the data from ADCxCR register after each conversion
Scan conversion	Repeatedly conversion of the specified channels, use “End of Scan” interrupt INT2 (after conversion of the end channel), read out the data from ADCxCD0...ADCxD31 registers

To set interrupts after conversion has been completed, see [44.7. ADC Interrupt Generation and DMA Access \(Page No.1065\)](#).

The conversion results are protected depending on the setting of the ADxCS2.INTE2 bit. Check [44.8. Protection of the ADC Results \(Page No.1066\)](#).

[Figure 44.6-4](#) shows the operation of A/D converter in Scan conversion mode. Channels AN0 to AN3 are enabled. Scan conversion uses the ADC Channel Data registers and the “End of Scan” interrupt INT2. Additionally, a conversion pause in Scan mode is shown.

Figure 44.6-4 A/D converter scan conversion with pause (channels 0 to 3 enabled)



- (1) Activation channel selection
- (2) A/D activation (Trigger: Software trigger/Reload timer/External trigger)
- (3) INT flag clear, PAUS flag clear

- (4) AN0 conversion
 - a. Sample time, conversion (comparision a + comparision b + comparision c)
 - b. Conversion end
 - c. Write the conversion value to result register of channel 0.
- (5) AN1 conversion
- (6) AN2 conversion
- (7) AN3 conversion
- (8) Next loop starts with AN0 conversion
- (9) INT2 (End of Scan) flag is set when end channel (AN3) is converted
- (9) AN0 conversion done. Because INT2 has not been cleared yet, the ADC protects the result register of AN0 against overwriting and enters PAUSE state.
- (10) INT2 and PAUS flags cleared by software, the ADC stores the result of AN0 and continues sampling AN1.

44.6.5 Stop Mode

In stop mode the analog input signal selected by the ANS bits and ANE bits are converted in order, but conversion operation pauses after each channel. The pause is released by applying another start signal.

At the completion of conversion on the end channel determined by the ANE bits, the converter returns to the ANS channel for analog input signal and repeats the conversion process continuously. When the start and end channel are the same (ANS=ANE), only a signal channel conversion is performed.

Examples:

- ANS=00000b, ANE=00011b

Start -> AN0 -> stop -> start -> AN1 -> stop -> start -> AN2 -> stop -> start -> AN3 -> stop -> start -> AN0 ... -> repeat

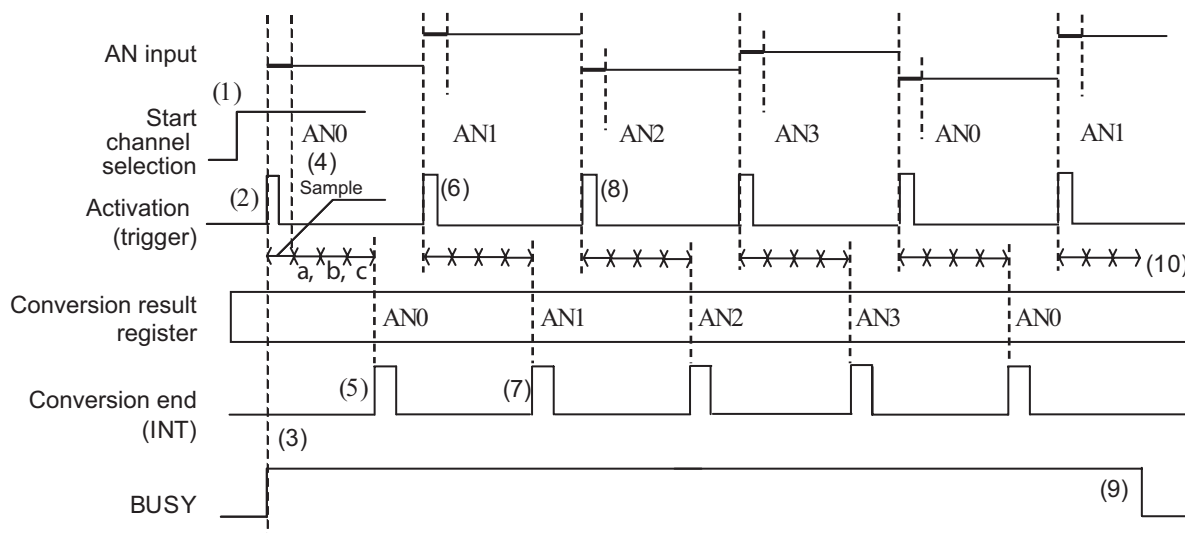
- ANS=00010b, ANE=00010b

Start -> AN2 -> stop -> start -> AN2 -> stop -> start -> AN2 ... -> repeat

In stop mode the startup source is the source determined by the STS1, STS0 bits. This mode enables synchronization of the conversion start signal.

Figure 44.6-5 shows the A/D converter operation in stop mode. In this example, channels AN0 to AN3 are enabled.

Figure 44.6-5 A/D Converter stop mode conversion



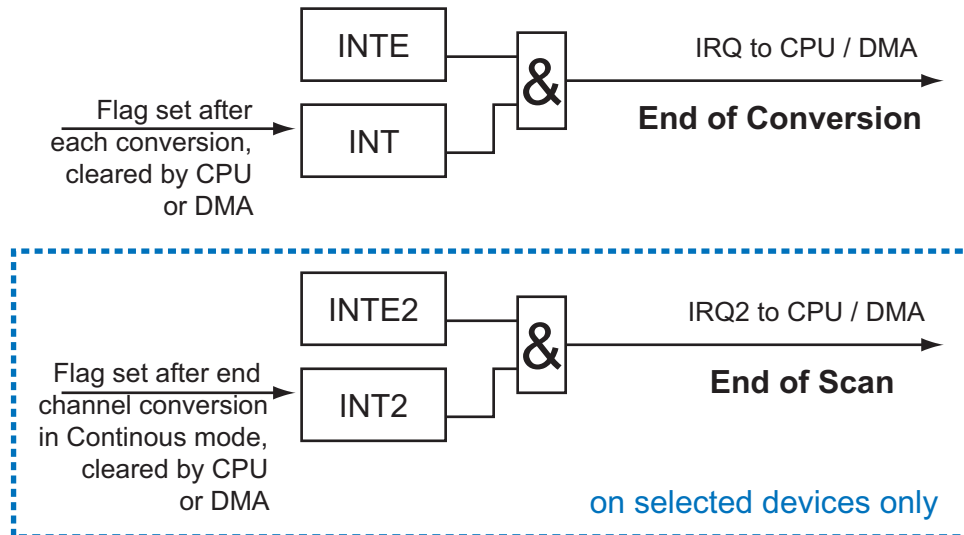
- (1) Activation channel selection
- (2) A/D activation (Trigger: Software trigger/Reload timer/External trigger)
- (3) BUSY flag set (the INT flag can be cleared also here)
- (4) AN0 conversion
 - a. Sampling time, conversion (comparison a + comparison b + comparison c)
 - b. Conversion end
 - c. Write the conversion value to result register.
- (5) INT flag set. INT is then cleared after reading out the conversion result register. The A/D converter waits for the next trigger.
- (6) A/D activation (Trigger) for channel AN1
- (7) INT flag set after AN1 conversion. The A/D converter waits for the next trigger.
- (8) A/D activation (Trigger) for the next channel
- (9) BUSY is forcibly cleared to stop the conversion
- (11) Conversion of channel AN1 is cancelled.

MB91460 Series

44.7. ADC Interrupt Generation and DMA Access

There are there are 2 ADC interrupt sources: End of Conversion and End of Scan. End of Conversion is available on all devices, whereas End of Scan can be used on selected devices only (see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#)).

Figure 44.7-1 A/D Converter interrupts



44.7.1 End of Conversion

The End of Conversion (EoC) interrupt is enabled by ADxCS1.INTE bit and is available on all devices of MB91460 series. If EoC is enabled, it appears after any conversion cycle. It is recommended to use DMA transfer to read out the data from ADCR.

44.7.2 End of Scan

The End of Scan (EoS) interrupt is enabled by ADxCS2.INTE2 bit. If EoS is enabled, it appears after the conversion of the end channel, which is defined by the setting of ADxECH register.

If the End of Conversion interrupt is enabled in parallel, both interrupt bits are set. In this case it is recommended that the interrupt routine reads out ADxCS2 register (containing mirrored bits of ADxCS1[7:4]) to check where the interrupt comes from.

44.7.3 ADC Interrupt Level registers and vectors

The ADC interrupts are connected to interrupt #134 and interrupt #135 as shown in [Table 44.7-1](#)

Table 44.7-1 ADC interrupt level registers and vectors

Interrupt	Interrupt number		Interrupt level		Interrupt vector		DMA	
	Decimal	Hexadecimal	Setting Register	Register address	Offset	Default Vector address	RN	Stop
A/D Converter 0 End of Conv	134	86	ICR59	47B _H	1E4 _H	000FFDE4 _H	14, 112	—
A/D Converter 0 End of Scan							120	—
A/D Converter 1 End of Conv	135	87	ICR59	47B _H	1E0 _H	000FFDE0 _H	113	—
A/D Converter 1 End of Scan							121	—

44.7.4 DMA Transfer

DMA transfer can be triggered by End of Conversion interrupt or by End of Scan interrupt. The interrupts are assigned to separate DMA resource numbers (please refer to the Interrupt Vector Table).

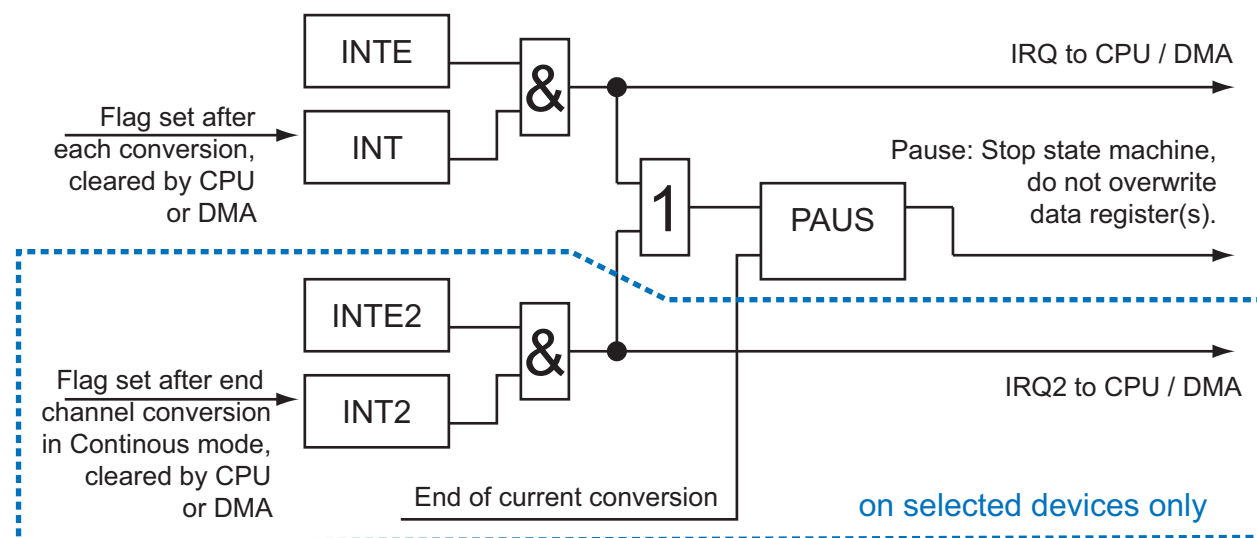
The automatic interrupt clear after DMA transfer works for End of Conversion and for End of Scan separately.

44.8. Protection of the ADC Results

This section explains the data register protection, which is implemented for the common data register ADxCR as well as the 32 separated ADC channel data registers ADCxD0 to ADCxD31 ($x = 0$ or 1 for ADC0, ADC1). Note that the separated ADC channel data registers are available on selected devices only, see [Table 44.1-1 A/D converter feature list \(Page No.1037\)](#).

If a conversion is finished and the data of the previous conversion has not been read out before, previous data would be overwritten. To avoid this problem, the conversion data is not stored in the data registers until the previous value has been read out (e.g. by DMA). A/D conversion halts during this time and the ADxCS1.PAUS flag is set. The ADC state machine is simply stopped. It resumes when both interrupt flags (ADxCS1.INT and ADxCS2.INT2) are cleared.

Figure 44.8-1 A/D converter data protection



44.8.1 Protection of ADxCR

If ADxCS1.INTE==1, PAUS is set when data of any channel is ready for writing to the registers, when IRQ (End of Conversion) is active. Because in this mode the protection function is active after each single conversion, the ADxCR register is protected (and the ADC channel data register is protected too).

After the CPU or DMA have read the data registers and cleared the INT flag, the conversion continues.

Note: Please refer to the description of the PAUS bit in section [44.3.4 A/D Control Status Register \(ADCS1/ADxCS1\) \(Page No.1044\)](#).

44.8.2 Protection of ADCxD0...31

In continuous mode with INTE==0 and INTE2==1, PAUS is set when data of the **start channel** (set by **ADxSCH**) is ready for writing to the registers, but IRQ2 (End of Scan interrupt) is already active.

Example: Start channel =4, end channel=7, continuous mode, AD0CS1.INTE=0, AD0CS2.INTE2=1

Start by CPU --> convert channel 4 + save data to ADC0D4,
convert channel 5 + save data to ADC0D5,
convert channel 6 + save data to ADC0D6,

convert channel 7 + save data to ADC0D7 ---> End of Scan interrupt (IRQ2),
convert channel 4 + set PAUS (protect ADC0D4 and the following).

After the CPU or DMA have read the data registers and cleared the INT2 flag, the scan conversion continues.

Note: Please refer to the description of the PAUS bit in section [44.3.4 A/D Control Status Register \(ADCS1/ADxCS1\)](#) (Page No.1044).

44.9. Setting

Table 44.9-1 Settings needed to use A/D - Single Shot Conversion Mode

Setting	Setting Registers	Setting Procedure*
Mode selection (Single shot conversion)	A/D control (ADxCS)	See 44.10.1
Bit length selection		See 44.10.2
Channel selection		See 44.10.4
Conversion time setting	Conversion time setting (ADCT)	See 44.10.3
To program the AN pin as an input	Port Function (PFR26-PFR29) and Extra Port Function (EPFR26-EPFR27)	See 44.10.5
A/D activation trigger selection	A/D control (ADxCS)	See 44.10.7
A/D activation trigger generation Software trigger -> Software trigger bit setting		See 44.10.8
Reload timer -> Reload timer rising output		
External trigger -> Inputs a trigger to the ADTG(0, 1) pin.		
Conversion end flag check		
Conversion value read	Data buffers (ADxCR)	See 44.10.10

*: For the setting procedure, refer to the section indicated by the number.

Table 44.9-2 Settings needed to use A/D - Continuous Conversion Mode

Setting	Setting Registers	Setting Procedure*
Mode selection (Continuous conversion)	A/D control (ADxCS)	See 44.10.1
Bit length selection		See 44.10.2
Starting channel selection		See 44.10.4
Conversion time setting	Conversion time setting (ADxCT)	See 44.10.3
Program the AN pin as an input.	Port Function (PFR26-PFR29) and Extra Port Function (EPFR26-EPFR27)	See 44.10.5
A/D activation trigger selection	A/D control (ADxCS)	See 44.10.7
A/D activation trigger generation Software trigger -> Software trigger bit setting		See 44.10.8
Reload timer -> Reload timer falling output		
External trigger -> Inputs a trigger to the ATGX pin.		
Conversion end flag check		
Conversion value read	Data buffers (ADxCR)	See 44.10.10

*: For the setting procedure, refer to the section indicated by the number.

Table 44.9-3 Forcing A/D operations to Stop

Setting	Setting Registers	Setting Procedure*
Forced stop	A/D control (ADxCS)	See 44.10.11

*: For the setting procedure, refer to the section indicated by the number.

Table 44.9-4 Items needed to enable A/D Interrupts

Setting	Setting Registers	Setting Procedure*
A/D interrupt vector and A/D interrupt level settings	See “ Chapter 24 Interrupt Control (Page No.429) ”.	See 44.10.12
A/D interrupt cause selection (A/D conversion end, A/D scan end)	A/D control registers (ADxCS)	See 44.10.13
A/D interrupt setting Clear interrupt requests. Enable interrupt requests.		See 44.10.14

*: For the setting procedure, refer to the section indicated by the number.

44.10. Q & A

44.10.1 What conversion modes are available and how are they selected?

The following modes of conversion are available:

- Single shot conversion mode, in which a specified sequence of channels are converted only once.
- Continuous conversion mode, in which a specified sequence of channels are converted in a loop.
 - In Continuous mode, the interrupt INT is used and the common data register ADxCR¹ is read.
 - Scan Conversion mode, is same as Continuous mode, but the interrupt INT2 (End of Scan) is used and the ADC channel result registers are read. Scan conversion is available only on selected devices, see section 44.1.1 [Features of the A/D Converter: \(Page No.1037\)](#).
- Stop conversion mode, in which a specified sequence of channels are converted, but conversion must be started after each channel.

Mode selection is made using the conversion mode selection bits (ADxCS0.MD[1:0]):

Table 44.10-1 A/D Conversion mode selection.

MD1	MD0	Operating mode
0	0	Single mode 1 (Reactivation during A/D conversion is allowed)
0	1	Single mode 2 (Reactivation during A/D conversion is not allowed)
1	0	Continuous mode or Scan conversion mode (Reactivation during A/D conversion is not allowed)
1	1	Stop mode (Reactivation during A/D conversion is not allowed)

44.10.2 How to specify a bit length (resolution)

Configure the conversion result storage bit resolution setting with ADxCS0.S10.

Table 44.10-2 A/D Converter resolution setting (Common result register ADxCR)

Operation Mode	ADxCS0.S10
To read conversion results from the ADxCR register with 10 bit	Set "0" [default].
To read conversion results from the ADxCR register with 8 bits ¹	Set "1".

1. The conversion result data is always written with 10 bits into ADxCR.
S10=1 multiplexes the read data ADxCR[9:2] --> R-Bus[7:0]

Table 44.10-3 A/D Converter resolution setting (Channel result registers ADCxDn)

Operation Mode	ADxCS0.S10
To write conversion results into the ADCxDn registers with 10 bit	Set "0" [default].
To write conversion results into the ADCxDn registers with 8 bits ¹	Set "1".

1. The conversion result data is written with 10 bits or 8 bits into ADCxRn.
S10=1 multiplexes the ADC macro output SAR[9:2] --> ADCxRn[7:0]
(x = 0,1 for ADC0, ADC1; n = 0 to 31 for ADC channel)

Note: The ADC channel data registers ADCxD0 to ADCxD31 are available on selected devcies only.

1. x= 0 or 1 for ADC0, ADC1

44.10.3 How to set a conversion time

Use Conversion Time Setting registers ADCT to set.

- **[bit 15 to 10] CT5-0 (A/D comparison time set)**
 - These bits specify clock division of comparison time.
 - Setting "000001" means one division (=CLKP).
 - It's not allowed to set these bits "000000".
 - Initialized these bits to "000100" by reset.
- Comparison time = CT value * CLKP cycle * 10 + (4 * CLKP)

Note: For details please see [44.5.3 Setting the Comparison Time \(Page No.1058\)](#).

- **[bit 9 to 0] ST9-0 (Analog input sampling time set)**
 - These bits specify sampling time of analog input.
 - Initialized these bits to "0000101100" by reset.
- Sampling time = ST value * CLKP cycle

Note: For details please see [44.5.2 Accuracy and Setting of the Sampling time \(Page No.1057\)](#).

44.10.4 How to enable the channels to be converted

In the A/D Converter macro, 3 registers control which channels are to be converted:

- The A/D Channel Setting registers (see [44.3.9 A/D Channel Setting Register \(ADSCH/ADxSCH, ADECH/ADxECH\) \(Page No.1053\)](#)) control which analog channel is the start channel and which one is the end channel in a loop.
- The A/D Input Enable register (see [44.3.2 A/D Input Enable Register \(ADER/ADxER\) \(Page No.1042\)](#)) controls which channels are converted between start- and end channel, and which channels are skipped.

44.10.5 How to enable analog pin input

For ADC channels AN0-AN31, there are two possibilities to set analog pins as an input. For both methods, the corresponding analog input channels must be enabled in the ADxERH,ADxERL registers. Then either one of the following two methods can be followed:

- **Method 1: Using Port Function register PFR and Extra Port Function register EPFR.**

Table 44.10-4 A/D Conversion analog channel enable with PFR/EPFR

Operation	PFR setting	EPFR setting
To program the AN0 pin as an input	PFR29.0 = '1'	EPFR29.0 = '0'
To program the AN1 pin as an input	PFR29.1 = '1'	EPFR29.1 = '0'
To program the AN2 pin as an input	PFR29.2 = '1'	EPFR29.2 = '0'
To program the AN3 pin as an input	PFR29.3 = '1'	EPFR29.3 = '0'
To program the AN4 pin as an input	PFR29.4 = '1'	EPFR29.4 = '0'
To program the AN5 pin as an input	PFR29.5 = '1'	EPFR29.5 = '0'
To program the AN6 pin as an input	PFR29.6 = '1'	EPFR29.6 = '0'
To program the AN7 pin as an input	PFR29.7 = '1'	EPFR29.7 = '0'
To program the AN8 pin as an input	PFR28.0 = '1'	EPFR28.0 = '0'
To program the AN9 pin as an input	PFR28.1 = '1'	EPFR28.1 = '0'
To program the AN10 pin as an input	PFR28.2 = '1'	EPFR28.2 = '0'

Operation	PFR setting	EPFR setting
To program the AN11 pin as an input	PFR28.3 = '1'	EPFR28.3 = '0'
To program the AN12 pin as an input	PFR28.4 = '1'	EPFR28.4 = '0'
To program the AN13 pin as an input	PFR28.5 = '1'	EPFR28.5 = '0'
To program the AN14 pin as an input * ¹	PFR28.6 = '1'	EPFR28.6 = '0'
To program the AN15 pin as an input * ¹	PFR28.7 = '1'	EPFR28.7 = '0'
To program the AN16 pin as an input	PFR27.0 = '1'	EPFR27.0 = '1'
To program the AN17 pin as an input	PFR27.1 = '1'	EPFR27.1 = '1'
To program the AN18 pin as an input	PFR27.2 = '1'	EPFR27.2 = '1'
To program the AN19 pin as an input	PFR27.3 = '1'	EPFR27.3 = '1'
To program the AN20 pin as an input	PFR27.4 = '1'	EPFR27.4 = '1'
To program the AN21 pin as an input	PFR27.5 = '1'	EPFR27.5 = '1'
To program the AN22 pin as an input	PFR27.6 = '1'	EPFR27.6 = '1'
To program the AN23 pin as an input	PFR27.7 = '1'	EPFR27.7 = '1'
To program the AN24 pin as an input	PFR26.0 = '1'	EPFR26.0 = '1'
To program the AN25 pin as an input	PFR26.1 = '1'	EPFR26.1 = '1'
To program the AN26 pin as an input	PFR26.2 = '1'	EPFR26.2 = '1'
To program the AN27 pin as an input	PFR26.3 = '1'	EPFR26.3 = '1'
To program the AN28 pin as an input	PFR26.4 = '1'	EPFR26.4 = '1'
To program the AN29 pin as an input	PFR26.5 = '1'	EPFR26.5 = '1'
To program the AN30 pin as an input	PFR26.6 = '1'	EPFR26.6 = '1'
To program the AN31 pin as an input	PFR26.7 = '1'	EPFR26.7 = '1'

1. Analogue input channels AN14 and AN15 can be also used as D/A converter outputs. In case of exclusive A/D converter usage do not enable the D/A converter output with DACR.DAE[1:0] since there is no priority of A/D converter input. See [Chapter 45 D/A Converter \(DAC\) \(Page No.1089\)](#) for further information about D/A converter.

• Method 2: Using Global ADC Analog Channel Enable (ADCHE).

Set ADCHE bit in the PORTEN register. This allows A/D conversion without PFR setting, also when the port is in digital output mode. This method is available on selected devices only, see section [44.1.1 Features of the A/D Converter: \(Page No.1037\)](#).

PORTEN Register Address: 0x0498 Access: Byte

7	6	5	4	3	2	1	0	Bit
-	-	-	-	-	ADCHE	CPORTEN	GPORTEN	
X	X	X	X	X	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	Attribute

● For ADC channels AN32-AN53:

For these analog channels, no PFR/EPFR setting is available and ADCHE is always considered as 1.

For details and limitations, please see section [44.4. Analog Input Connections \(Page No.1054\)](#).

44.10.6 How to enable analog pin input for Stepper Motor Controller

Use Port Function register PFR and Extra Port Function register EPFR.

Operation	PFR setting	EPFR setting
To program the AN16 pin as an input for SMC1P0	PFR27.0 = '1'	EPFR27.0 = '0'
To program the AN17 pin as an input for SMC1M0	PFR27.1 = '1'	EPFR27.1 = '0'
To program the AN18 pin as an input for SMC2P0	PFR27.2 = '1'	EPFR27.2 = '0'
To program the AN19 pin as an input for SMC2M0	PFR27.3 = '1'	EPFR27.3 = '0'
To program the AN20 pin as an input for SMC1P1	PFR27.4 = '1'	EPFR27.4 = '0'
To program the AN21 pin as an input for SMC1M1	PFR27.5 = '1'	EPFR27.5 = '0'
To program the AN22 pin as an input for SMC2P1	PFR27.6 = '1'	EPFR27.6 = '0'
To program the AN23 pin as an input for SMC2M1	PFR27.7 = '1'	EPFR27.7 = '0'
To program the AN24 pin as an input for SMC1P2	PFR26.0 = '1'	EPFR26.0 = '0'
To program the AN25 pin as an input for SMC1M2	PFR26.1 = '1'	EPFR26.1 = '0'
To program the AN26 pin as an input for SMC2P2	PFR26.2 = '1'	EPFR26.2 = '0'
To program the AN27 pin as an input for SMC2M2	PFR26.3 = '1'	EPFR26.3 = '0'
To program the AN28 pin as an input for SMC1P3	PFR26.4 = '1'	EPFR26.4 = '0'
To program the AN29 pin as an input for SMC1M3	PFR26.5 = '1'	EPFR26.5 = '0'
To program the AN30 pin as an input for SMC2P3	PFR26.6 = '1'	EPFR26.6 = '0'
To program the AN31 pin as an input for SMC2M3	PFR26.7 = '1'	EPFR26.7 = '0'

See [Chapter 43 Stepper Motor Controller \(SMC\) \(Page No.1025\)](#) about using the A/D converter for SMC operation.

44.10.7 Possibilities to activate the A/D converter

There are three types of activation triggers:

- Software trigger
- Reload timer rising signal
- External trigger input falling signal

To set an activation trigger, use Activation Trigger Selection bits (ADxCS1.STS[1: 0]).

A/D Activation Trigger	Activation Trigger Selection bit (STS[1: 0])
To specify a software trigger	Set "00".
To specify an external trigger/software trigger	Set "01".
To specify a reload timer/software trigger	Set "10".
To specify an external trigger/reload timer/software trigger	Set "11".

The converter A/D is activated on the first instance of any one of these causes selected.

44.10.8 How to activate the A/D converter

- **Generating a software trigger**

A software trigger is generated using A/D Conversion Software Trigger bits (ADxCS1.STRT).

Operation	A/D Conversion Software Trigger Bit (STRT)
To generate a software trigger	Write "1".

- **Activating A/D converter with reload timer 7**

The reload timers must be setup and activated. For more information, see [Chapter 38 Reload Timer \(RLT\) \(Page No.921\)](#). When an underflow of a reload timer causes the reload timer output signal to rise, an activation trigger is generated.

- **Activating A/D converter with an external trigger**

Use external trigger input pin ATGX to generate an external trigger.

The external trigger input pin is set using Port Function bits (PFR16.7) and Extra Port Function Register (EPFR16.7).

Operation	Setting
To program the ATGX pin as a trigger input	Set PFR16.7 = '0' and DDR16.7 = '0' (port input mode)
	Set PFR16.7 = '1' and EPFR16.7 = '1' (function mode)

44.10.9 How to verify the end of a conversion

There are two ways to verify the end of a conversion, as follows:

- **Checking the A/D Conversion End of Conversion bit (ADxCS1.INT)**

(INT)	Description
If the read value is "0"	No A/D conversion end interrupt request
If the read value is "1"	A/D conversion end interrupt request

- **Checking the Operation Verification bits (ADxCS1.BUSY)**

(BUSY)	Setting
If the read value is "0"	A/D conversion end (stop)
If the read value is "1"	A/D conversion in progress

- **Checking the A/D Conversion End of Scan (ADxCS2.INT2)**

(INT2)	Description
If the read value is "0"	No A/D conversion end interrupt request
If the read value is "1"	A/D conversion end interrupt request

Note: If the End of Conversion interrupt is enabled in parallel, both interrupt bits are set. In this case it is recommended that the interrupt routine reads out ADxCS2 register (containing mirrored bits of ADxCS1[7:4]) to check where the interrupt comes from.

44.10.10 How to read a conversion value

The conversion value can be read from the Common Data Buffer register ADxCR. On selected devices, dedicated channel data registers ADCxD0-ADCxD31 are present. In this case, the conversion value is available on both the ADxCR and ADCxDn registers. Channels 0-31 are present on ADC0 and Channels 32-54 on ADC1. Refer to [44.3. Registers of the A/D Converter and Range Comparator \(Page No.1040\)](#) for addresses and register names.

44.10.11 How to force an A/D conversion operation to a stop

Use the Forced Stop bits (ADxCS1.BUSY)

Operation	Forced Stop Bit (BUSY)
To force an A/D conversion operation to a stop	Write "0".

The operation of the A/D is unaffected by writing "1" to the Forced Stop bit (BUSY).

44.10.12 What interrupt registers are used?

A/D interrupt vector, A/D interrupt level setting

The table below summarizes the relationships among the machine cycle, A/D number, interrupt level, and interrupt vector.

For more information about the interrupt level and interrupt vector, see [Chapter 24 Interrupt Control \(Page No.429\)](#).

	Interrupt Vector (Default)	Interrupt Level Setting Bit (ICR[4:0])
ADC0 INT ADC0 INT2 ¹	#134 Address: 0FFDE4 _H	Interrupt Level register (ICR59) Address: 0ADC 147B _H
ADC1 ² INT ADC1 INT2	#135 Address: 0FFDE0 _H	

1. INT2 is available on selected devcies only. The lines INT and INT2 are OR-connected.
2. ADC1 is available on selected devcies only.

44.10.13 What interrupts are available?

A/D Conversion End interrupt only. No interrupt cause selection bit is available.

Interrupt	Description / Notes
INT	End of Conversion
INT2 ¹	End of Scan

1. INT2 is available on selected devcies only.

44.10.14 How to enable, disable, clear interrupts

Interrupt Request Enable bit, Interrupt flag

Interrupts are enabled using the Interrupt Request Enable bits (ADxCS1.INTE, ADxCS2.INTE2).

INT, INT2	Interrupt Request Enable Bit (INTE, INTE2)
To disable interrupt requests	Set "0".
To enable interrupt requests	Set "1".

Interrupts are cleared using the Interrupt flags (ADxCS1.INT, ADxCS2.INT2).

INT, INT2	Interrupt Flag (INT)
To clear interrupt	Write "0" (latest when the conversion start trigger is set)

44.11. Caution

- Power-on sequence
It is necessary to turn on the MCU power (V_{DD}^*) before turning on the power to the A/D converter (AV_{CC5} , AV_{RH5}) and applying a voltage to the analog input.
- Input impedance of the analog input pin
The A/D converter has a built-in sample & hold circuit to receive the voltage present on the analog input pin in the sample hold capacitor after the activation of an A/D conversion. Therefore, if the analog input external circuit has a high output impedance, it may happen that analog input voltage fails to get stabilized within the sampling cycle. For this reason, keep the output impedance of the external circuit sufficiently low.
If the output impedance of the external circuit cannot be kept sufficiently low, lengthen the sampling time fully.
- As $AV_{RH}-AV_{SS}$ decreases the error grows in proportion.

MB91460 Series

44.12. Range Comparator Overview

The Range Comparator compares the A/D conversion result with 2 thresholds. There are 4 comparators with an upper and a lower threshold. The 32 ADC channels can be enabled for range comparison and assigned to one of the 4 comparators individually.

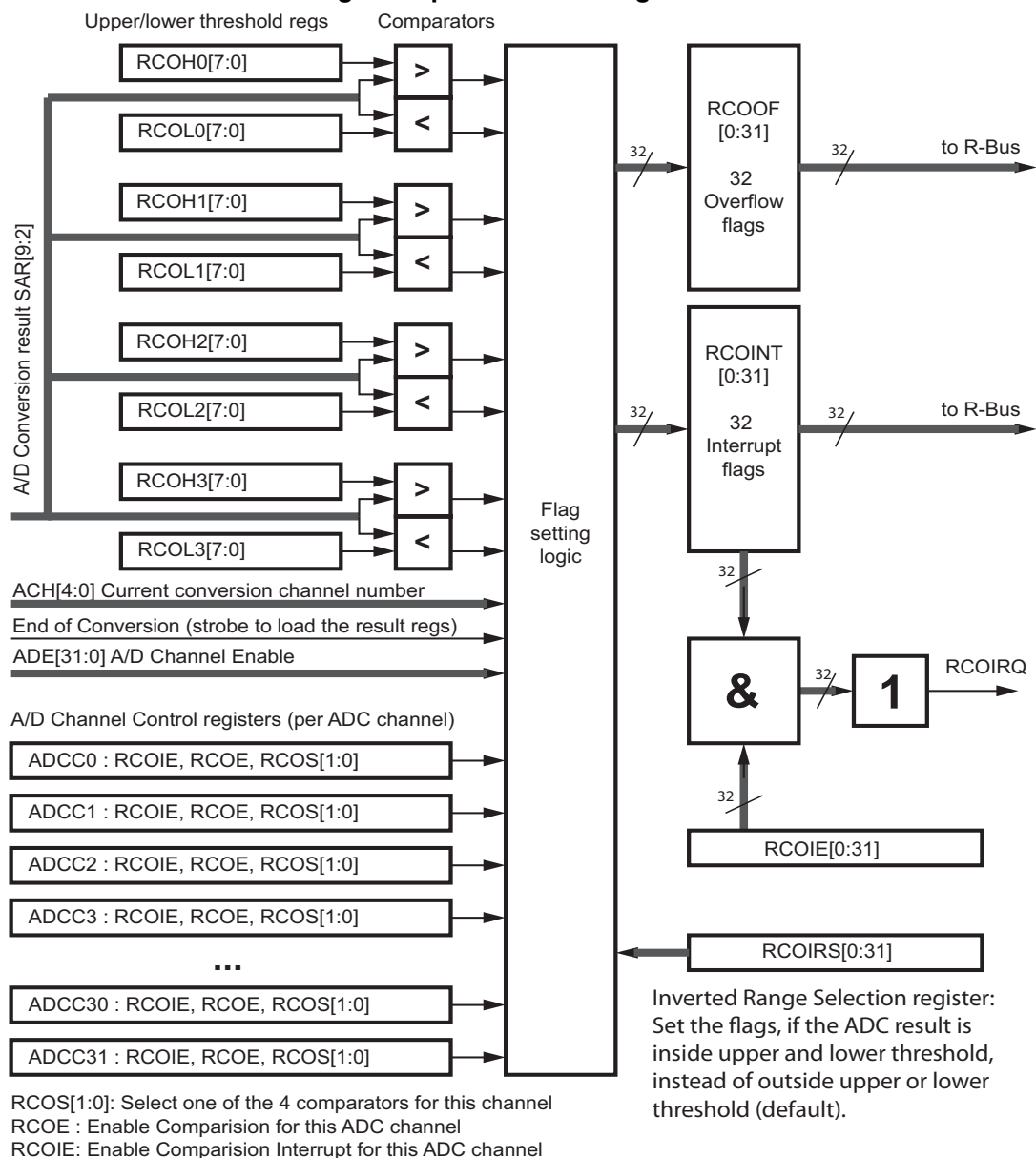
44.12.1 Range Comparator Structure

If enabled, the comparison will set up to 2 flags for this ADC channel:

- An interrupt flag RCOINT, signalling that the ADC result is outside the range or, by “inverted” configuration, inside the range.
- An overflow flag RCOOF, showing that the range violation was an overflow and no underflow.

Furthermore, each ADC channel can be enabled to send an interrupt request to the CPU, if the RCOINT flag is set. Figure 44.12-1 shows the block diagram:

Figure 44.12-1 A/D Converter Range Comparator block diagram



44.13. Range Comparator Registers

The Range Comparator (RCO) has the following registers per A/D Converter macro:

- RCOxHn[7:0] : Upper threshold register, one register per comparator block (n = 0...3)
- RCOxLn[7:0] : Lower threshold register, one register per comparator block (n = 0...3)
- ADxCCm[7:0] : ADC channel control, one register per 2 ADC channels (m = 0...15)
- RCOxIRS[0:31] : RCO Inverted Range Selection, one bit per ADC channel
- RCOxOF[0:31] : RCO Overflow Flags, one bit per ADC channel, read-only
- RCOxINT[0:31] : RCO Interrupt Flags, one bit per ADC channel

Note: x=0 or 1 for ADC0 and ADC1

44.13.1 Range Comparator Threshold registers (RCOxH0/L0 to RCOxH3/L3)

- **RCOxH0-3** : Higher threshold, access: Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

15	14	13	12	11	10	9	8	Bit
RCOH7	RCOH6	RCOH5	RCOH4	RCOH3	RCOH2	RCOH1	RCOH0	
1	1	1	1	1	1	1	1	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\).](#))

- **[bit 7:0] RCOH[7:0] (Range Comparator High threshold)**

The RCOH bits define the higher comparison threshold of the Range Comparator channel.

The upper Range Comparator compares that the upper 8 bits of the ADC conversion result are higher than RCOH[7:0]

- **RCOxL0-3** : Lower threshold, access: Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

7	6	5	4	3	2	1	0	Bit
RCOL7	RCOL6	RCOL5	RCOL4	RCOL3	RCOL2	RCOL1	RCOL0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\).](#))

- **[bit 7:0] RCOL[7:0] (Range Comparator Low threshold)**

The RCOL bits define the lower comparison threshold of the Range Comparator channel.

The lower Range Comparator compares that the upper 8 bits of the ADC conversion result are lower than RCOL[7:0]

44.13.2 A/D Converter Channel Control registers (ADxCC0 to ADxCC15)

The A/D channel control registers serve 2 ADC channels per register and control the range comparison for these channels.

ADxCC0 register controls A/D channels 0 + 1,

ADxCC1 register controls A/D channels 2 + 3,

...

ADxCC15 register controls A/D channels 30 + 31

- **ADxCC0-15:** Access: Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

7	6	5	4	3	2	1	0	Bit
RCOIE1	RCOE1	RCOS11	RCOS10	RCOIE0	RCOE0	RCOS01	RCOS00	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
Bits 7:4 control A/D channels 1,3,5,7,...31				Bits 3:0 control A/D channels 0,2,4,6,...,30				

(For the attributes, refer to section 1.5.4 Meaning of Bit Attribute Symbols (Page No.15).)

- **[bit 7,3] RCOIE1, RCOIE0 (Range Comparator Interrupt enable)**

The RCOIE bits enable the Range Comparator interrupt for the corresponding ADC channel.

RCOIE1 RCOIE0	Function
0	RCO interrupt for this ADC channel is disabled [default]
1	RCO interrupt for this ADC channel is enabled

- **[bit 6,2] RCOE1, RCOE0 (Range Comparator operation enable)**

The RCOE bits enable the Range Comparison for the corresponding ADC channel:

RCOE1 RCOE0	Function
0	RCO disabled, RCO flags for this ADC channel will not be set [default]
1	RCO enabled for this ADC channel

- **[bits 5:4,1:0] RCOS1[1:0], RCOS0[1:0] (converter channel select)**

These bits select the Range Comparator channel to be assigned to this A/D converter channel:

RCOS01, RCOS00 RCOS11, RCOS10	Function
00	Select range comparator channel 0 for this ADC channel [default]
01	Select range comparator channel 1 for this ADC channel
10	Select range comparator channel 2 for this ADC channel
11	Select range comparator channel 3 for this ADC channel

44.13.3 Inverted Range Selection register (RCOxIRS[0:31])

The RCOxIRS register controls that the comparison should check for “out of range” or “inside range”.

The 32 bits of RCOIRS are organized “per ADC channel”. ADC channel 0 is located on the MSB of the register and ADC channel 31 is on the LSB.

- **RCOxIRS** : Access: Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

31	30	29	28	27	26	25	24	Bit
RCOIRS0	RCOIRS1	RCOIRS2	RCOIRS3	RCOIRS4	RCOIRS5	RCOIRS6	RCOIRS7	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
23	22	21	20	19	18	17	16	Bit
RCOIRS8	RCOIRS9	RCOIRS10	RCOIRS11	RCOIRS12	RCOIRS13	RCOIRS14	RCOIRS15	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
15	14	13	12	11	10	9	8	Bit
RCOIRS16	RCOIRS17	RCOIRS18	RCOIRS19	RCOIRS20	RCOIRS21	RCOIRS22	RCOIRS23	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	Bit
RCOIRS24	RCOIRS25	RCOIRS26	RCOIRS27	RCOIRS28	RCOIRS29	RCOIRS30	RCOIRS31	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\).](#))

Note that bit[31] is assigned to ADC channel AN0, bit[30] is assigned to ADC channel AN1 and so on.

- **[bits 31:0] RCOIRS[0:31] (Inverted Range Select)**

The RCOIRS bits control how the Range Comparator result flags are set:

RCOIRS _n	Function
0 [default]	The range comparator flags are set when the ADC result is above the upper threshold OR below the lower threshold. That is called “ out of range ” mode.
1	The range comparator flags are set when the ADC result is below or equal the upper threshold AND above or equal the lower threshold. That is called “ inside range ” mode.

44.13.4 Range Comparator Interrupt Flags (RCOxINT[0:31])

The result of range comparison is stored in 2 flag registers:

- RCOxINT[0:31]: Range comparison interrupt flags
- RCOxOF[0:31]: Range comparison overflow flags

The Range Comparator Result flags are organized “per ADC channel”. There are 32 Range Comparator overflow flags and 32 interrupt flags. In case of a RCO interrupt, all interrupt flags can be read out by one 32-bit read operation and analyzed using the Bit Search Unit (see [Chapter 28 Bit Search \(Page No.521\)](#)). The Bit Search Unit will return the number of the interrupting channel. Since bit search works from MSB to LSB (from left to right), ADC channel 0 is located on the MSB of the registers and ADC channel 31 is on LSB.

- **RCOxINT[0:31]** : Access: Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

31	30	29	28	27	26	259	24	Bit
RCOINT0	RCOINT1	RCOINT2	RCOINT3	RCOINT4	RCOINT5	RCOINT6	RCOINT7	
0	0	0	0	0	0	0	0	Initial value
R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	Attribute
23	22	21	20	19	18	17	16	Bit
RCOINT8	RCOINT9	RCOINT10	RCOINT11	RCOINT12	RCOINT13	RCOINT14	RCOINT15	
0	0	0	0	0	0	0	0	Initial value
R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	Attribute
15	14	13	12	11	10	9	8	Bit
RCOINT16	RCOINT17	RCOINT18	RCOINT19	RCOINT20	RCOINT21	RCOINT22	RCOINT23	
0	0	0	0	0	0	0	0	Initial value
R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	Attribute
7	6	5	4	3	2	1	0	Bit
RCOINT24	RCOINT25	RCOINT26	RCOINT27	RCOINT28	RCOINT29	RCOINT30	RCOINT31	
0	0	0	0	0	0	0	0	Initial value
R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	Attribute

(For the attributes, refer to section [1.5.4 Meaning of Bit Attribute Symbols \(Page No.15\)](#).)

Note that bit[31] is assigned to ADC channel 0, bit[30] is assigned to ADC channel one and so on.

- **[bits 31:0] RCOINT[0:31] (Range Comparator Interrupt flags)**

The RCOINT flags show that a “out of range” or “inside range” condition has been found on the ADC channel.

- The bits are set under the following condition:

- the ADC channel is enabled ADxER.ADE[n] is set and
- the range comparison for this channel is enabled ADxCCn.RCOE[i] is set and
- the conversion of the ADC channel is just finished and
- an interrupt condition was found (see [Table 44.13-1](#) on next page).

- The bits are cleared by writing 0 or by software reset (RST). Writing 1 has no effect.
- Read-modify-write operations read 1.

The interrupt condition depends on the comparison results and the RCOxIRS setting for this channel:

Table 44.13-1 Range Comparator interrupt conditions

RCOIRS	Upper threshold comparator	Lower threshold comparator	Interrupt condition	RCOINT flag is...	RCOOF flag is...
0 (out of range mode)	1	x	INT condition: above range	set	set
	0	0	-	-	-
	x	1	INT condition: below range	set	cleared
1 (inside range mode)	1	x	-	-	-
	0	0	INT condition: inside range	set	cleared
	x	1	-	-	-

Note: The upper threshold comparator returns 1 if the upper 8 bits of the ADC result are greater than the threshold value in RCOH[7:0].

The lower threshold comparator returns 1 if the upper 8 bits of the ADC result are smaller than the threshold value in RCOL[7:0].

44.13.5 Range Comparator Overflow Flags (RCOxOF[0:31])

The Range Comparator Overflow flags store the output of the upper threshold comparator at the time the RCOINT flag is set the first time (e.g. at the rising edge of RCOINT).

- **RCOxOF[0:31]** : Access: Read-only, Word, Half-word, Byte (x=0 or 1 for ADC0 and ADC1)

31	30	29	28	27	26	25	24	Bit
RCOOF0	RCOOF1	RCOOF2	RCOOF3	RCOOF4	RCOOF5	RCOOF6	RCOOF7	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute
23	22	21	20	19	18	17	16	Bit
RCOOF8	RCOOF9	RCOOF10	RCOOF11	RCOOF12	RCOOF13	RCOOF14	RCOOF15	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute
15	14	13	12	11	10	9	8	Bit
RCOOF16	RCOOF17	RCOOF18	RCOOF19	RCOOF20	RCOOF21	RCOOF22	RCOOF23	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute
7	6	5	4	3	2	1	0	Bit
RCOOF24	RCOOF25	RCOOF26	RCOOF27	RCOOF28	RCOOF29	RCOOF30	RCOOF31	
0	0	0	0	0	0	0	0	Initial value
R	R	R	R	R	R	R	R	Attribute

Note that bit[31] is assigned to ADC channel 0, bit[30] is assigned to ADC channel one and so on.

- [bits 31:0] RCOOF[0:31] (Range Comparator Overflow flag)

The RCOOF read-only flags store the output signal of the upper threshold comparator at the time when an interrupt condition (see Table 44.13-1 above) appeared and the corresponding RCOINT flag was **not** set before. So the RCOOF flags indicate the upper comparator state when the RCOINT flag had the rising edge.

The RCOOF flag for a ADC channel is loaded with the upper threshold comparator output signal under the following condition:

- the corresponding RCOINT flag is not yet set and
- the corresponding RCOINT flag has a set condition in this cycle.

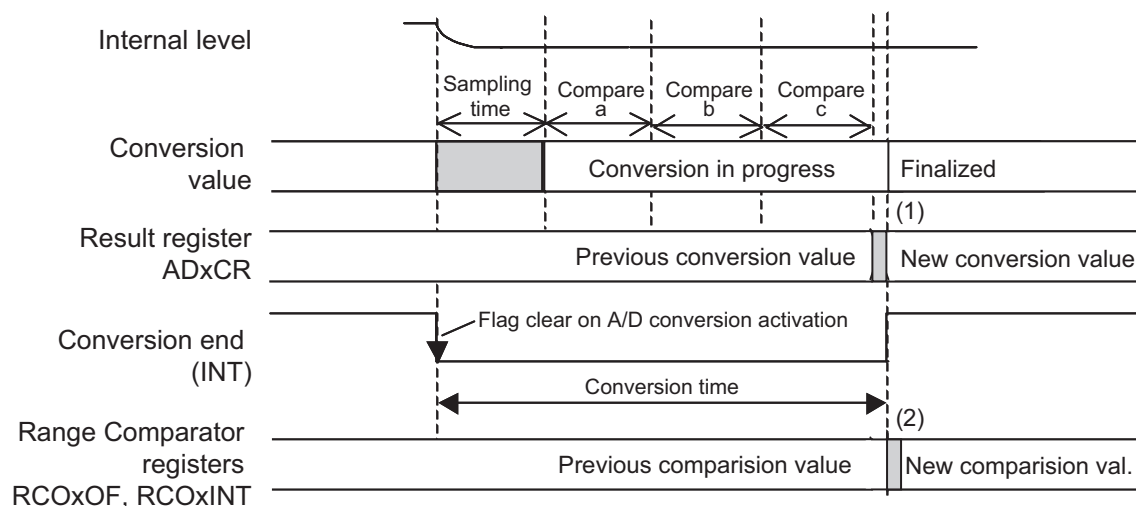
The flags are initialized by software reset (RST).

RCOOFn	Function
0 [default]	The output of the upper threshold comparator was 0
1	The output of the upper threshold comparator was 1

44.14. Range Comparator Operation

If enabled, the Range Comparator becomes active in the cycle when the ADC has finished a conversion and the result data is stored in the ADC result register ADxCR. The comparator result flags are valid one CLKP cycle later.

Figure 44.14-1 ADC Range Comparator flag setting



(1) The A/D conversion is done and the ADC result register is loaded.

(2) The Range Comparator flags are set 1 CLKP cycle later.

44.14.1 How to Setup the Range Comparator

The Range Comparator (RCO) can be enabled or disabled anytime. But it is recommended to stop the ADC operation when RCO control registers are changed, just to avoid wrong range comparison results.

The following example shows how to enable the Range Comparison.

- Condition: $AV_{CC5} = AV_{RH5} = 5.0V$, ADC in 10-bit mode, 5.0V is value 1022 approximately.
- Upper RCO threshold should be 4.0V, lower RCO threshold should be 1.0V.
- ADC channels 3,4 and 5 are to be compared to the thresholds mentioned above using RCO channel 1.
- ADC channels 3 and 4 are to be compared for “outside range”, while channel 5 should be compared for “inside range”. All 3 channels should issue RCO interrupt.
- Assume that the ADC is already converting channels 0 to 7 in continuous mode.

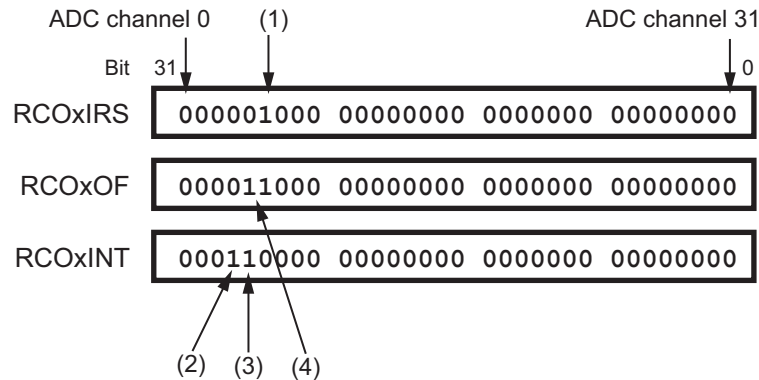
Table 44.14-1 Range Comparator Setup Example

Step	Software	Explanation
1	// Set the thresholds RCO0H1=818; // 1022*4V/5V RCO0L1=204; // 1022*1V/5V	RCO0: ADC channels 3-5 are on macro ADC0. Therefore we have to use Range Comparator macro RCO0. RCO0H1 is upper threshold of RCO0 channel 1. RCO0L1 is lower threshold of RCO0 channel 1.
2	AD0CS1_BUSY=0;	Forcibly terminate the ADC conversion (optional)
3	RCO0IRS=0x04000000	Inverted Range Select register, bit 5 (from left to right) is set to enable ADC channel 5 “inside range” comparison.
4	RCO0INT=0x00000000;	Clear the interrupt flags before enabling the RCO.
5.1	AD0CC1=0b1101_0000	ADC channel 3: RCOIE=1 (INT enable), RCOE=1 (RCO enable), RCOS[1:0]=01 (select RCO channel 1) ADC channel 2: RCO unused.
5.2	AD0CC2=0b1101_1101	ADC channel 5: RCOIE=1, RCOE=1, RCOS=01 ADC channel 4: RCOIE=1, RCOE=1, RCOS=01 (same as above)
6	ICR30=11;	Interrupt level for RCO0
7	AD0CS1_BUSY=1; AD0CS1_STRT=1;	Restart the ADC conversion (if stopped in step 2). The RCO flags are set after the next conversion of the channel.

44.14.2 How to Obtain Range Comparator Results

The Range Comparator result registers RCOxINT and RCOxOF (x = 0 or 1 for RCO0, RCO1) can be read by the CPU. It is possible to poll RCOxINT, but it is easier to use the RCO interrupt.

From our example setup above, the CPU may read the following values from the RCO registers:



- Note that the ADC channels are located from left to right in the registers.
- (1) RCOxIRS is listed here to remember that channel 5 should compare for “inside range”.
- (2) ADC channel 3 was converted first, and its interrupt flag caused the interrupt request to the CPU.
- (3) Assume that there was a certain interrupt response time and channel 4 and 5 have been converted meanwhile. The flag of channel 4 is set, the flag of channel 5 not.
- (4) The overflow flags of channel 4 and 5 are set.

The values in RCOxINT and RCOxOF give us the following result

- Channel 3 is “out of range” and has no overflow: The value was below the lower threshold.
- Channel 4 is “out of range” and has overflow: The value was above the upper threshold.
- Channel 5 is not “inside range” (no INT flag) and has overflow: The value was above the upper threshold.

■ Using the Bit Search Unit to determine the interrupting channel

The Bit Search Unit can be used to determine which ADC channel caused a RCO interrupt.

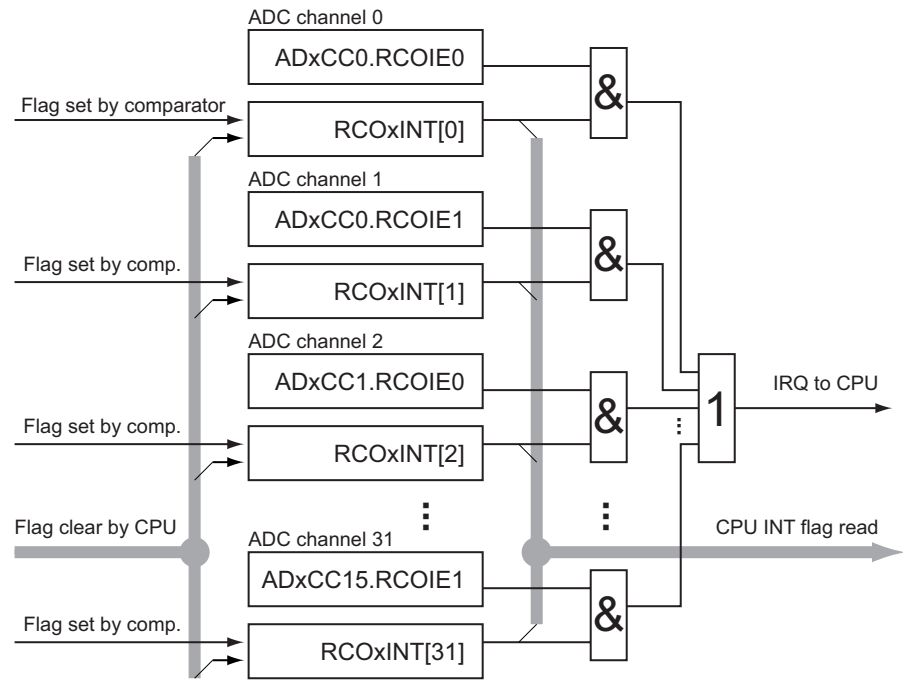
The Bit Search Unit (see [Chapter 28 Bit Search \(Page No.521\)](#)) is used to detect the first ‘0’ position, the first ‘1’ position or the first changing position for data written in specific registers.

- Read the value of RCOxINT register and write it into Bit Search BSD1 register.
- The Bit Search will check for the first ‘1’ bit from left to right and returns its position:
 - If bit[31] is ‘1’ then it returns value 0 for ADC channel 0.
 - Our example above returns value 3 for AC channel 3.
 - If bit[0] is ‘1’ then it returns value 31 for ADC channel 31.
 - If no bit is ‘1’ then it returns value 32.

44.15. Range Comparator Interrupt

The Range Comparator has one interrupt output line RCOxIRQ per RCO module. The interrupt output line becomes active if at least one of the Range Comparator interrupt flags RCOINT[31:0] is set and the corresponding interrupt enable bit in the ADxCC register is set. (x = 0 or 1 for RCO0 and RCO1)

Figure 44.15-1 Range Comparator interrupt structure



It is not possible to activate a DMA request from the range comparator interrupts.
The RCO interrupts are connected to interrupt #77 and interrupt #81 as shown in Table 44.15-1.

Table 44.15-1 Range Comparator interrupt level registers and vectors

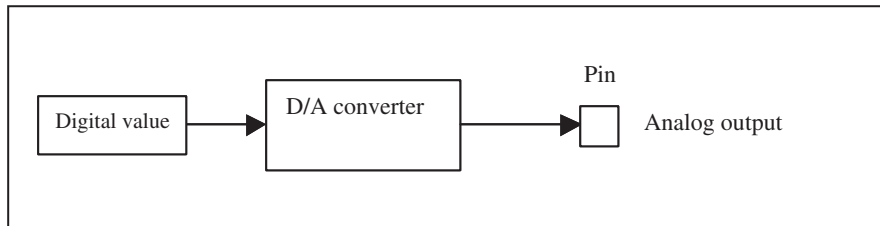
Interrupt	Interrupt number		Interrupt level		Interrupt vector		DMA	
	Decimal	Hexadecimal	Setting Register	Register address	Offset	Default Vector address	RN	Stop
LIN-USART 8 RX / APIX [®] Event / Fatal Error	76	4C	ICR30	45E _H	2CC _H	000FFECC _H	64 / —	64 / —
LIN-USART 8 TX / LIN-USART 8 EOT ADC 0 Range Comparator	77	4D			2C8 _H	000FFEC8 _H	65 / —	—
...
LIN-USART 10 RX / APIX [®] Transaction Buffer	80	50	ICR32	460 _H	2BC _H	000FFEBC _H	68 / 160-175	68 / 160-175
LIN-USART 10 TX LIN-USART 10 EOT ADC 1 Range Comparator	81	51			2B8 _H	000FFEB8 _H	69 / —	—

MB91460 Series**Chapter 45 D/A Converter (DAC)**

45.1. Overview

The D/A converter converts digital values to an analog output signal using an R-2R type conversion.

Figure 45.1-1 Block diagram of D/A converter

**45.2. Features**

- Method : R-2R type conversion
 Quantity : 2 (Output: DA0 pin and DA1 pin)

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Conversion time : 0.45us (Typ) (Load capacitance = 20pF)
 2.0us (Typ) (Load capacitance = 100pF)
 Resolution : 10-bit resolution
 Output range : From AV_{SS} (0V) to $1023/1024 \times AV_{CC}$
 Interrupt : None
 Others : Power-down feature available (fixed 0 V output).
 Useful for saving current consumption when in SLEEP state.

45.3. Configuration

Figure 45.3-1 Configuration Diagram of D/A converter

D/A converter (0-1)

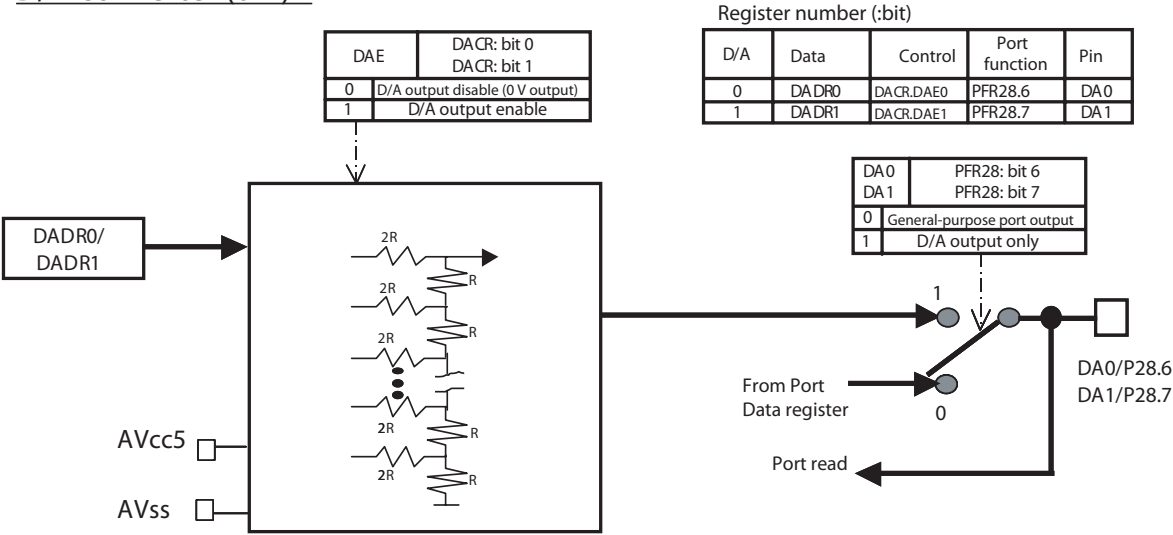


Figure 45.3-2 Register List

D/A Converter 0

Address	Bit	7	6	5	4	3	2	1	0		
000364H		---	---	---	---	---	---	DA9	DA8	DADR0	(D/A data 0)
000365H		DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	DADR0	(D/A data 0)
000361H		---	---	---	---	---	MD08	DAE1	DAE0	DACR	(D/A control)
000D9CH		DA1	DA0	---	---	---	---	---	---	PFR28	(Port function 28)

D/A Converter 1

Address	Bit	7	6	5	4	3	2	1	0		
000366H		---	---	---	---	---	---	DA9	DA8	DADR1	(D/A data 1)
000367H		DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	DADR1	(D/A data 1)
000361H		---	---	---	---	---	MD08	DAE1	DAE0	DACR	(D/A control)
000D9CH		DA1	DA0	---	---	---	---	---	---	PFR28	(Port function 28)

MB91460 Series

45.4. Registers

45.4.1 DADR: D/A Data Register

The D/A Data Register sets the output voltage of the D/A converter.

- **DADR0(ch0): Address 0364_H (Access: Byte, Half-word)**
- **DADR1(ch1): Address 0366_H (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	Bit
-	-	-	-	-	-	DA9	DA8	
-	-	-	-	-	-	X	X	Initial value
R0/W0	R0/W0	R0/W0	R0/W0	R0/W0	R0/W0	R/W	R/W	Attributes
7	6	5	4	3	2	1	0	Bit
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	
X	X	X	X	X	X	X	X	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

(For the attributes, refer to the “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- The D/A Data Register is not initialized on a reset.
- The setting is “000_H” - “3FF_H”.
- Read access returns DADR[9:0] in 10-bit mode and (DADR[7:0]<2) in 8-bit mode

45.4.2 DACR: D/A Control Register

The D/A Control Register enables/disables both DAC channels, and determines the resolution.

- **DACR(ch0/ch1): Address 0361_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
-	-	-	-	-	MD08	DAE1	DAE0	
-	-	-	-	-	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	Attributes

(For the attributes, refer to the “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit7-3: Undefined
At write, always write “0”. At read, the read value is indeterminate.
- bit0: D/A output control

DAE0	Operation
0	D/A output disabled
1	D/A output enabled

- Enables a converted analog level to be output from the DA pin.
(To place the DA0 pin in the output state, it is necessary to set PFR28.6=“1”.
- The D/A output equals 0.0 V when the D/A output control bit is “0”.
- bit1: D/A output control

DAE1	Operation
0	D/A output disabled
1	D/A output enabled

- Enables a converted analog level to be output from the DA pin.
(To place the DA1 pin in the output state, it is necessary to set PFR28.7=“1”.

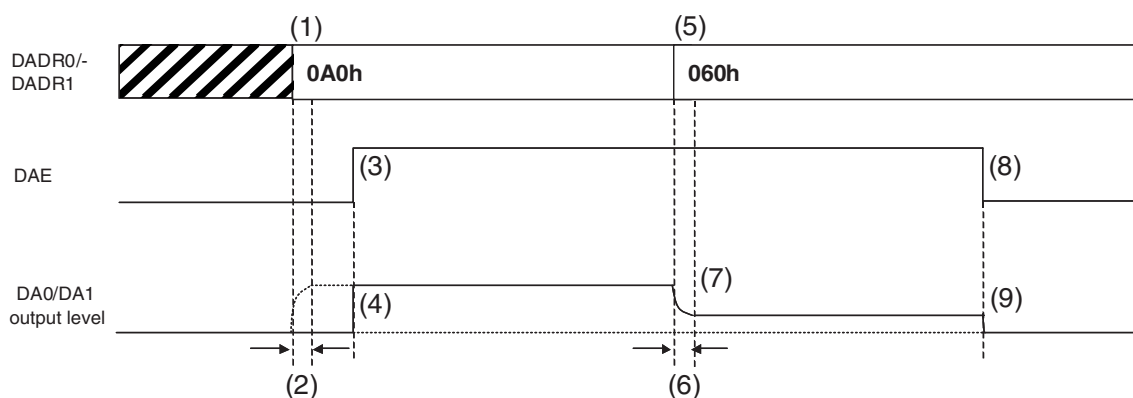
- The D/A output equals 0.0 V when the D/A output control bit is “0”.
- bit2: D/A 8-/10-bit mode control

MD08	Operation
0	D/A resolution is 10 bits, DADR[9:0] is converted
1	D/A resolution is 8 bits, (DADR[7:0] << 2) is converted

- In case MD08='1' the 8-bit value of DA7-DA0 (DADR[7:0]<<2) is output.

45.5. Operation

Figure 45.5-1 The operations of the D/A converter



- (1) Digital value setting (software-programmable)
- (2) D/A conversion in progress
- (3) Output enabled (software-programmable)
- (4) Analog value output
- (5) Digital value rewrite (software-programmable)
- (6) D/A conversion in progress
- (7) Output level finalized
- (8) Output disabled (software-programmable)
- (9) Fixed 0 V output

MB91460 Series**45.6. Setting****Table 45.6-1 Settings Needed to Use D/A**

Setting	Setting Registers	Setting Procedure*
Digital value settings	D/A Data Registers (DADR)	See 45.7.1
Pin settings	Port Function Register (PFR28.7, PFR28.6)	See 45.7.2
Output enabled	D/A Control Registers (DACR)	See 45.7.3

*:For the setting procedure, refer to the section indicated by the number.

Table 45.6-2 Settings Needed to Stop D/A Output

Setting	Setting Registers	Setting Procedure*
Output halted	D/A Control Registers (DACR)	See 45.7.3

*:For the setting procedure, refer to the section indicated by the number.

45.7. Q & A

45.7.1 How to set the resolution

Write digital values to the D/A Data Registers (DADR[7:0] for 8-bit mode, DADR[9:0] for 10-bit mode).

Access in a byte or halfword format.

D/A conversion begins immediately on writing.

45.7.2 How to program the D/A pins for D/A output?

DA Pin output setting

Setting is accomplished by writing "1" to the output specification bits (PFR28.7 for DA1), (PFR28.6 for DA0).

(Switch the port to DA pin output by software programming.)

Pins	Control bit location	
DA0 pin	PFR28.6 = '1'	DA0 Output specification bit (DA0)
DA1 pin	PFR28.7 = '1'	DA1 Output specification bit (DA1)

45.7.3 How to enable or disable D/A output?

Use the D/A output control bits (DACR.DAE0), (DACR.DAE1).

Operations	D/A output control bit (DAE)
To disable output	Set "0".
To enable output	Set "1".

0 V (= AVss) is output when disabled. This is functional even during stop mode.

45.7.4 How to activate a D/A conversion?

A conversion begins on writing a digital value. See [45.7.1](#)

45.7.5 What is the formula used to work out the value necessary to produce an expected voltage?

Equation

$$\text{Value} = \{V (\text{Expected analog value}) \times 1024\} / (AV_{CC})$$

To output 2.8 V from the pin with $AV_{CC} = 5.0V$, for example.

$$(2.8V \times 1024) / 5.0V = 573.44 \Rightarrow \text{Value} = 573$$

45.8. Caution

- The table below lists the output voltages of the D/A converter (in 10-bit resolution mode).

Table 45.8-1 D/A conversion output voltages in 10-bit mode

DADR Settings	D/A Converter Output Voltage Value
000 _H	0V (AV _{SS} =0.0V)
001 _H	$1/1024 \times AV_{CC}$ V
002 _H	$2/1024 \times AV_{CC}$ V
~	~
3FD _H	$1021/1024 \times AV_{CC}$ V
3FE _H	$1022/1024 \times AV_{CC}$ V
3FF _H	$1023/1024 \times AV_{CC}$ V
When stopped	0V (AV _{SS} =0.0V)

- The table below lists the output voltages of the D/A converter (in 8-bit resolution mode).

Table 45.8-2 D/A conversion output voltages in 8-bit mode

DADR Settings	D/A Converter Output Voltage Value
00 _H	0V (AV _{SS} =0.0V)
01 _H	$1/256 \times AV_{CC}$ V
02 _H	$2/256 \times AV_{CC}$ V
~	~
FD _H	$253/256 \times AV_{CC}$ V
FE _H	$254/256 \times AV_{CC}$ V
FF _H	$255/256 \times AV_{CC}$ V
When stopped	0V (AV _{SS} =0.0V)

- The conversion speed depends on the line load capacitance.

Indicates the conversion speed of the D/A converter.

Table 45.8-3 D/A conversion speed

Load Capacitance	Conversion Speed (TYP)
20pF	0.45μs
100pF	2.0μs

- Power supply
 - The power supply of the analog circuit in the D /A converter is AV_{CC5}.

MB91460 Series

Chapter 46 Alarm Comparator

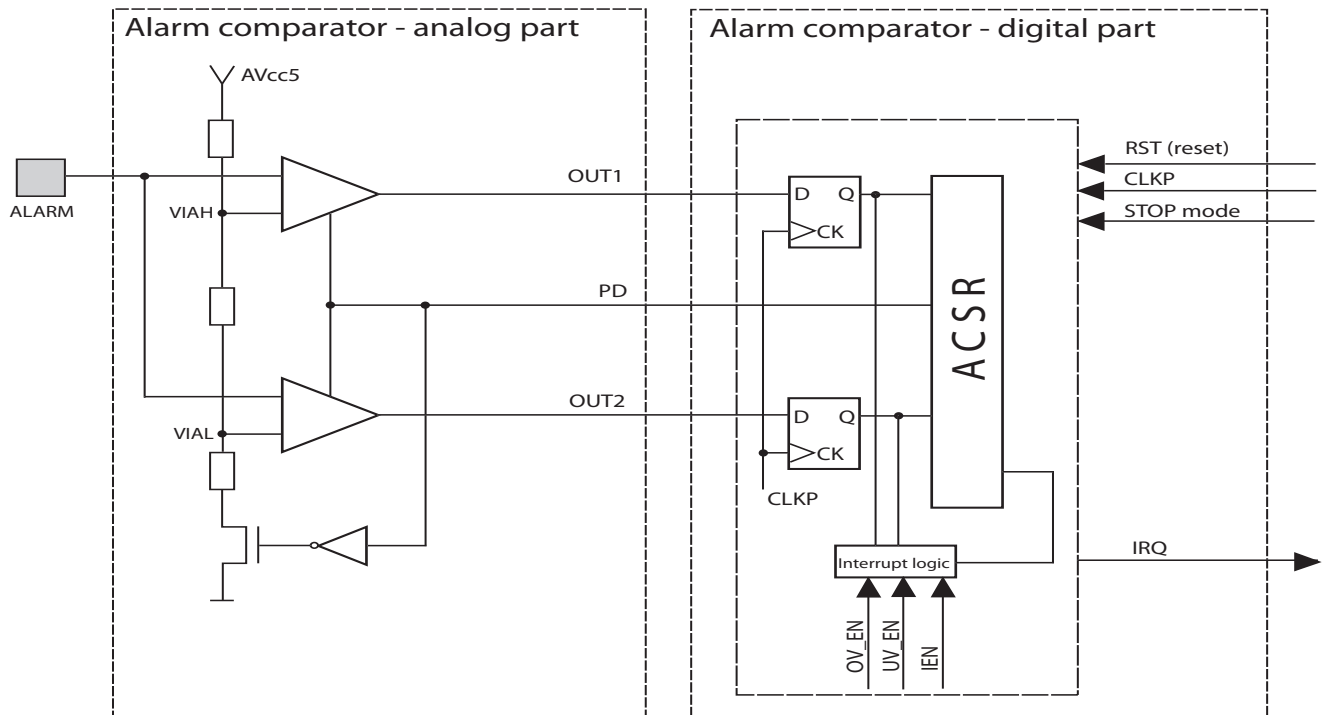
46.1. Overview

This chapter provides an overview of the Alarm Comparator (also called Under/Overvoltage Detection), describes the register structure and functions, and the operation of the Alarm Comparator.

Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

46.2. Block Diagram

Figure 46.2-1 Block diagram of the Alarm Comparator



46.3. Alarm Comparator Control/Status Register (ACSR)

- **ACSR0 (ch0): Address 01AD_H (Access: Byte)**
- **ACSR1 (ch1): Address 01AF_H (Access: Byte)**

Figure 46.3-1 Alarm Comparator Control/Status Register (ACSR0, ACSR1)

		Bits								Initial value
	Address	7	6	5	4	3	2	1	0	
ACSR0	0000 01AD _H									011XXX00 _B
ACSR1	0000 01AF _H	MD	OV_EN	UV_EN	OUT2	OUT1	IRQ	IEN	PD	↔ Access
		R/W	R/W	R/W	R	R	R/W	R/W	R/W	

Figure 46.3-2 Structure of Alarm comparator control/status register

Bit 7: MD Mode select

1	Fast mode enabled (tCOMP _F)
0	Slow mode enabled (tCOM _P _S) [Initial value]

Bit 6: OV_EN Overvoltage Enable

1	Interrupt enabled in case of overvoltage. [Initial value]
0	No interrupt in case of overvoltage

Bit 5: UV_EN Undervoltage Enable

1	Interrupt enabled in case of undervoltage. [Initial value]
0	No interrupt in case of undervoltage.

Bit 4: OUT2 synchronized output of Alarm comparator UV output.

0	analog input voltage < 2/5 AV _{cc5}
1	analog input voltage > 2/5 AV _{cc5}

Bit 3: OUT1 synchronized output of Alarm comparator OV output.

1	analog input voltage > 4/5 AV _{cc5}
0	analog input voltage < 4/5 AV _{cc5}

MB91460 Series**Bit 2: IRQ** Interrupt flag.

1	Under- or overvoltage condition detected
0	No under- or overvoltage condition detected.

Bit 1: IEN Interrupt request enable bit.

1	Interrupt request enabled
0	Interrupt request disabled [Initial value]

Bit 0: PD Power down bit.

1	Power down
0	Runmode [Initial value]

46.4. Operation Modes

The alarm comparator circuit can operate in interrupt or polling mode. The internal interrupt logic will detect each interrupt event independent from setting of the IEN bit.

46.4.1 Interrupt Mode (IEN=1)

The following truth table describes the valid interrupt events

Table 46.4-1 Valid interrupt events

OUT2	OUT1	IRQ	analog input voltage range
1	1	1	Input pin voltage $V_{in} > V_{IAH}$ (overvoltage)
1	0	0	$V_{IAL} < \text{Input pin voltage} < V_{IAH}$ (normal operation)
0	0	1	Input pin voltage $V_{in} < V_{IAL}$ (undervoltage)

The interrupt flag (IRQ) will be set with the next positive transition of CLKP after detecting an interrupt event. If IEN=1 this will create an interrupt request to the CPU. In order to determine the reason for the asserted interrupt - if both interrupts are enabled - it is necessary to read the ACSR register immediately inside the interrupt service routine. OUT2 and OUT1 always contain the actual status of the comparator outputs, i.e. the interrupt trigger event will not be stored.

46.4.2 Polling Mode (IEN=0)

The Interrupt flag (IRQ) will be set by an active interrupt event and can be reset by writing to the ACSR register. The ACSR can be polled continuously in order to monitor the input voltage which is feed to the AC comparator inputs.

46.4.3 Setting and Resetting of IRQ flag

The IRQ flag of the ACSR register can be reset to zero by writing a "0" to it. Writing a "1" to the IRQ flag of ACSR register has no effect. IRQ can only be set to "1" by hardware, i.e. by the outputs of the

comparator circuits. IRQ will remain active as long as an active interrupt status is detected, even if a “0” is written to it.

A bitset command performed on the ACSR register will result in a RMW access on the R-Bus. Every read access during performing a RMW command will return a “1” for the IRQ flag to the CPU. That avoids any loss in detecting interrupt events due to software setting of IRQ flag.

46.4.4 Power Down Modes of the Alarm Comparator

The alarm comparator circuit has the following power down modes:

Table 46.4-2 Alarm Comparator power down modes

STOP mode	PD bit	analog part
0	0	Run mode
0	1	Power down mode
1	0	Power down mode
1	1	Power down mode

Warning: The outputs of the alarm comparator (analog part) will remain undefined for at least 3 us after power on and also after reentering the run mode. Therefore it is strongly recommended to reset the interrupt flag before the interrupts are enabled.

MB91460 Series

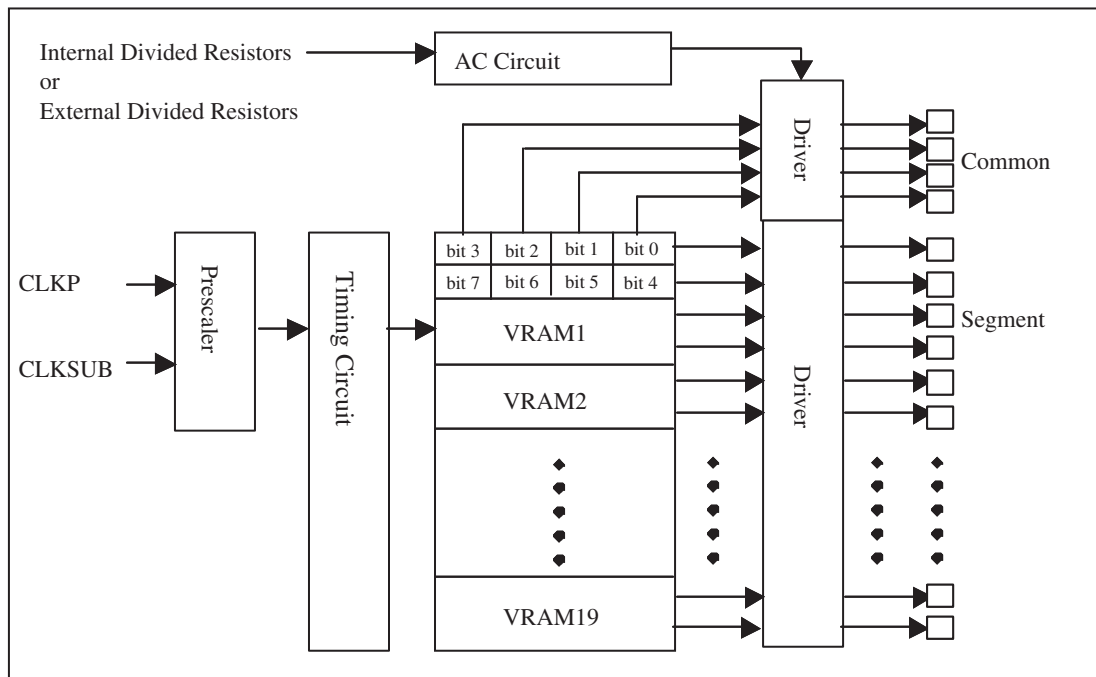
Chapter 47 LCD Controller

47.1. Overview

LCD allows display of up to 160 cells and selection of a duty cycle from 1/2, 1/3 and 1/4.

LCD has many applications.

Figure 47.1-1 Block Diagram of LCD Controller



47.2. Features

- Quantity: 1 (4 common x 40 segment)

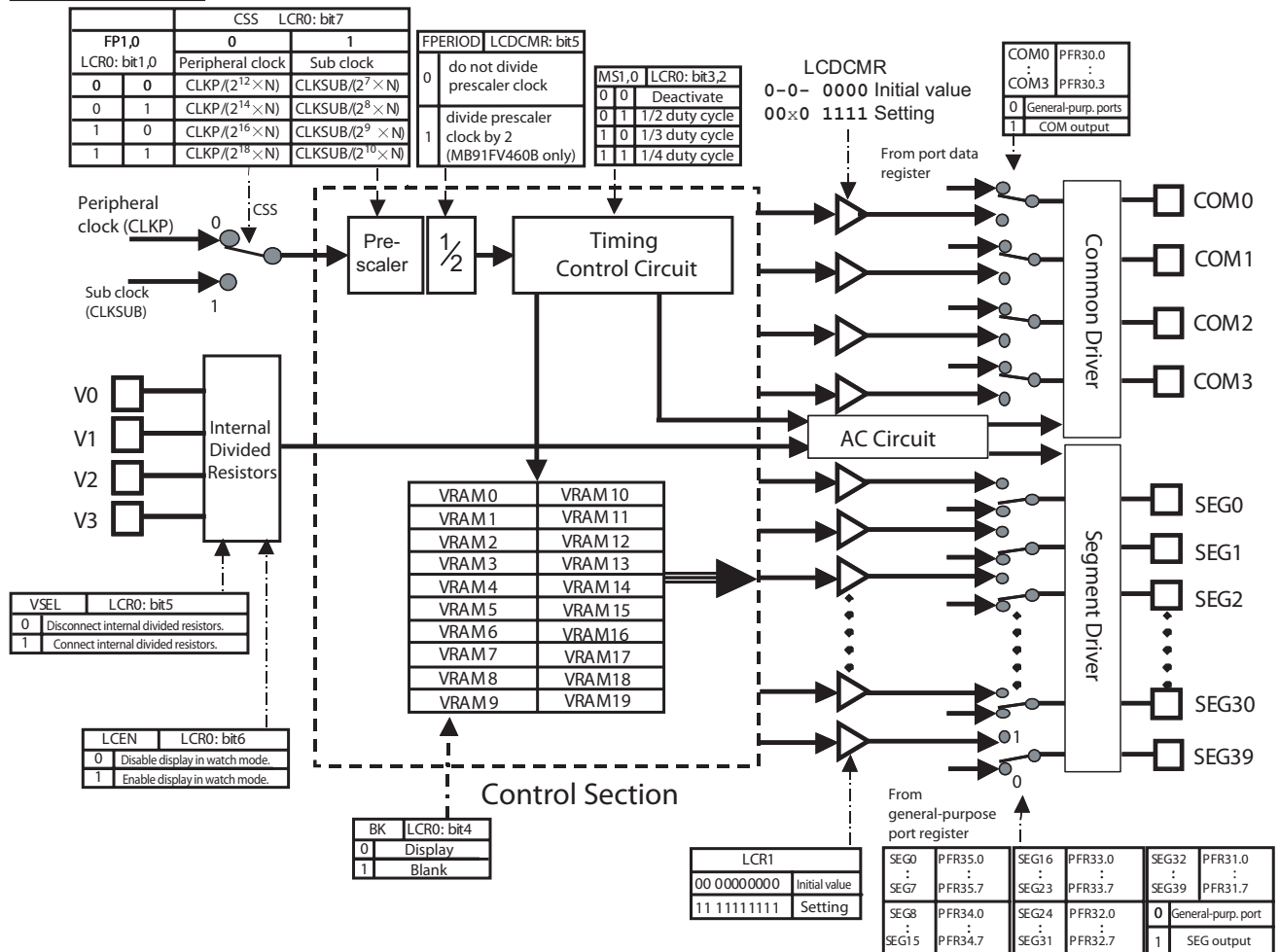
Note: To find out which resources are implemented on a device, please refer to section [2.3.3 MB91460 Series Resource Lineup \(Page No.29\)](#).

- Display: Up to 160 cells (for 1/4 duty cycle)
- Duty cycle: Selectable from options: 1/2, 1/3 and 1/4.
- Bias: Fixed at 1/3
- Frame period: Clock source is selectable from CLKP and CLKSUB (Four prescaler options for each clock source are possible)
- Driver: Built-in (for internal divided resistors), or external divided resistors can be connected to the V0 - V3 pins.
- Data memory: Built-in 20-byte data memory for display
- STOP state: Enable LCD display in the Sub-STOP state (MB91FV460B only).
- Blank display: Selectable.
- Pin: The SEG0-39 of COM0-4 pin usage can be switched between general and specialized purposes.
- Other: External divided resistors can be also used to shut off the current when LCD is deactivated.

47.3. Configuration

Figure 47.3-1 Configuration Diagram of LCD Controller

LCD Controller



Note: For details on ports, refer to “Chapter 30 I/O Ports (Page No.565)” and “Chapter 3 MB91460 Series Basic Information (Page No.35)”.

Figure 47.3-2 Register List

LCDC Controller

Address	Bit	7	6	5	4	3	2	1	0		
0000E8H		DTCH	---	FPERIOD	---	COMEN3	COMEN2	COMEN1	COMEN0	LCDCMR	(Common pin switching)
0000E9H		CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0	LCR0	(LCDC control 0)
0000EAH		---	---	---	---	---	---	SEGEN9	SEGEN8	LCR1H	(LCDC control 1 high)
0000EBH		SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0	LCR1L	(LCDC control 1 low)
0000ECH		SEG1	SEG1	SEG1	SEG1	SEG0	SEG0	SEG0	SEG0	VRAM0	(Data memory for display)
0000EDH		SEG3	SEG3	SEG3	SEG3	SEG2	SEG2	SEG2	SEG2	VRAM1	
0000EEH		SEG5	SEG5	SEG5	SEG5	SEG4	SEG4	SEG4	SEG4	VRAM2	
0000EFH		SEG7	SEG7	SEG7	SEG7	SEG6	SEG6	SEG6	SEG6	VRAM3	
0000F0H		SEG9	SEG9	SEG9	SEG9	SEG8	SEG8	SEG8	SEG8	VRAM4	
0000F1H		SEG11	SEG11	SEG11	SEG11	SEG10	SEG10	SEG10	SEG10	VRAM5	
0000F2H		SEG13	SEG13	SEG13	SEG13	SEG12	SEG12	SEG12	SEG12	VRAM6	
0000F3H		SEG15	SEG15	SEG15	SEG15	SEG14	SEG14	SEG14	SEG14	VRAM7	
0000F4H		SEG17	SEG17	SEG17	SEG17	SEG16	SEG16	SEG16	SEG16	VRAM8	
0000F5H		SEG19	SEG19	SEG19	SEG19	SEG18	SEG18	SEG18	SEG18	VRAM9	
0000F6H		SEG21	SEG21	SEG21	SEG21	SEG20	SEG20	SEG20	SEG20	VRAM10	
0000F7H		SEG23	SEG23	SEG23	SEG23	SEG22	SEG22	SEG22	SEG22	VRAM11	
0000F8H		SEG25	SEG25	SEG25	SEG25	SEG24	SEG24	SEG24	SEG24	VRAM12	
0000F9H		SEG27	SEG27	SEG27	SEG27	SEG26	SEG26	SEG26	SEG26	VRAM13	
0000FAH		SEG29	SEG29	SEG29	SEG29	SEG28	SEG28	SEG28	SEG28	VRAM14	
0000FBH		SEG31	SEG31	SEG31	SEG31	SEG30	SEG30	SEG30	SEG30	VRAM15	
0000FCH		SEG33	SEG33	SEG33	SEG33	SEG32	SEG32	SEG32	SEG32	VRAM16	
0000FDH		SEG35	SEG35	SEG35	SEG35	SEG34	SEG34	SEG34	SEG34	VRAM17	
0000FEH		SEG37	SEG37	SEG37	SEG37	SEG36	SEG36	SEG36	SEG36	VRAM18	
0000FFH		SEG39	SEG39	SEG39	SEG39	SEG38	SEG38	SEG38	SEG38	VRAM19	
000D9EH		V3	V2	V1	V0	COM3	COM2	COM1	COM0	PFR30	(Port function)
000D9FH		SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32	PFR31	
000DA0H		SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24	PFR32	
000DA1H		SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	PFR33	
000DA2H		SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	PFR34	
000DA3H		SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	PFR35	

47.4. Registers

47.4.1 LCR0: LCDC Control Register 0

This register is used to select a frame period and its clock and the display mode, to enable/disable LCD display and the operation in the watch mode, and to control the drive power source.

- **LCR0: Address 0E9_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
CSS	LCEN	VSEL	BK	MS1	MS0	FP1	FP0	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” chapter.)

- bit7: Select the frame period generation clock

CSS	Operation
0	Peripheral clock (CLKP)
1	Sub clock (Sub oscillator X0A or RC-Clock depending on bit CSVCR.SCKS)

When the peripheral clock is selected, LCD does not operate if the Main clock stops (that is, LCD is in the Sub clock mode in which the Main clock stops, or in the Sub-STOP state).

- bit6: Enable operation in Sub-STOP

LCEN	Operation
0	Disable LCD display in the Sub-STOP state.
1	Enable LCD display in the Sub-STOP state.

To enable LCD display, the frame period generation clock select bit (CSS) must be also set to “1”.

- bit5: Control LCD drive power supply

VSEL	Operation
0	Disconnect internal divided resistors.
1	Connect internal divided resistors.

To connect external divided resistors, the LCD drive power supply control bit (VSEL) must be set to “0”.

- bit4: Select blanking

BK	Operation
0	Enable LCD display.
1	Disable (blank) LCD display.

- bit3-2: Select a display mode

MS1	MS0	Display mode
0	0	Deactivate LCD.
0	1	1/2 duty cycle output mode (Time division number: N=4, COM0-COM1)
1	0	1/3 duty cycle output mode (Time division number: N=6, COM0-COM2)
1	1	1/4 duty cycle output mode (Time division number: N=8, COM0-COM3)

If the display mode select bit (MS[1:0]) is set to “00”, LCD Controller ceases to operate.

A “L” level is output through common/segment pins.

The time division number N describes the number of LCD output pulses in a frame. For example, with 1/4 duty cycle, the frame consists of 8 output pulses: A sequence of 4 pulses + the same sequence inverted.

MB91460 Series

- bit1-0: Frame period

FP1	FP0	Frame period		
		When peripheral clock is selected, covering F_{CLKP} frequencies from 1 MHz to 64 MHz	When Sub clock is selected, covering $CLKSUB$ frequencies from 32 kHz to 200 kHz	
0	0	$CLKP/(2^{13} \times N)$	$CLKSUB/(2^3 \times N)$	MB91V460A
0	1	$CLKP/(2^{14} \times N)$	$CLKSUB/(2^4 \times N)$	
1	0	$CLKP/(2^{15} \times N)$	$CLKSUB/(2^5 \times N)$	
1	1	$CLKP/(2^{16} \times N)$	$CLKSUB/(2^6 \times N)$	
0	0	$CLKP/(2^{12} \times N)$	$CLKSUB/(2^7 \times N)$	MB91FV460B
0	1	$CLKP/(2^{14} \times N)$	$CLKSUB/(2^8 \times N)$	
1	0	$CLKP/(2^{16} \times N)$	$CLKSUB/(2^9 \times N)$	
1	1	$CLKP/(2^{18} \times N)$	$CLKSUB/(2^{10} \times N)$	

F_{CLKP} Peripheral clock (CLKP) frequency

$CLKSUB$ Sub clock frequency (X0A: 32 to 100kHz, RC-Clock: 50 to 200 kHz)

N Time division number (Selected with the display mode select bits, MS1 and MS0.)

Note: If LCDCMR:FPERIOD is '1' the frame period is elongated by factor 2 (MB91FV460B only).

Select an appropriate value in accordance with the frame frequency of the LCD panel.

Note: The LCD can also be operated on RC-clock. This can be enabled by setting bit CSCFG.CSC3 (switches to RC clock instead of 32kHz Sub Clock). Please see also section [13.4.5 CSCFG: Clock Source Configuration Register \(Page No.299\)](#).

47.4.2 Frame Period Setting Examples (MB91FV460B)

The following table gives some examples for setting the frame period.

Clock	Freq	MS[1:0]	FP[1:0]	Frame Frequency	Notes
CLKP	2 MHz	01	01	30.5 Hz	
		10	00	81 Hz	
		11	00	61 Hz	
CLKP	32 MHz	01	11	30.5 Hz	
		10	10	81 Hz	
		11	10	61 Hz	
CLKP	50 MHz	01	11	48 Hz	
		10	11	32 Hz	
		11	10	95 Hz	
X0A	32 kHz	01	00	64 Hz	
			01	32 Hz	
		10	00	42 Hz	
		11	00	32 Hz	

Clock	Freq	MS[1:0]	FP[1:0]	Frame Frequency	Notes
RC Clock	100 kHz *1	01	01 10	97 Hz 48 Hz	
		10	01 10	64 Hz 32 Hz	
		11	01	48 Hz	

1. RC clock frequency is min. 50 kHz, typ 100 kHz, max. 200 kHz

MB91460 Series

47.4.3 VRAM: Data Memory for Display

Memory area (VRAM) for setting display data

- **VRAM0 (SEG0, SEG1): Address 0EC_H (Access: Byte, Halfword, Word)**
- **VRAM1 (SEG2, SEG3): Address 0ED_H (Access: Byte, Halfword, Word)**
- **VRAM2 (SEG4, SEG5): Address 0EE_H (Access: Byte, Halfword, Word)**
- **VRAM3 (SEG6, SEG7): Address 0EF_H (Access: Byte, Halfword, Word)**
- **VRAM4 (SEG8, SEG9): Address 0F0_H (Access: Byte, Halfword, Word)**
- **VRAM5 (SEG10, SEG11): Address 0F1_H (Access: Byte, Halfword, Word)**
- **VRAM6 (SEG12, SEG13): Address 0F2_H (Access: Byte, Halfword, Word)**
- **VRAM7 (SEG14, SEG15): Address 0F3_H (Access: Byte, Halfword, Word)**
- **VRAM8 (SEG16, SEG17): Address 0F4_H (Access: Byte, Halfword, Word)**
- **VRAM9 (SEG18, SEG19): Address 0F5_H (Access: Byte, Halfword, Word)**
- **VRAM10 (SEG20, SEG21): Address 0F6_H (Access: Byte, Halfword, Word)**
- **VRAM11 (SEG22, SEG23): Address 0F7_H (Access: Byte, Halfword, Word)**
- **VRAM12 (SEG24, SEG25): Address 0F8_H (Access: Byte, Halfword, Word)**
- **VRAM13 (SEG26, SEG27): Address 0F9_H (Access: Byte, Halfword, Word)**
- **VRAM14 (SEG28, SEG29): Address 0FA_H (Access: Byte, Halfword, Word)**
- **VRAM15 (SEG30, SEG31): Address 0FB_H (Access: Byte, Halfword, Word)**
- **VRAM16 (SEG32, SEG33): Address 0FC_H (Access: Byte, Halfword, Word)**
- **VRAM17 (SEG34, SEG35): Address 0FD_H (Access: Byte, Halfword, Word)**
- **VRAM18 (SEG36, SEG37): Address 0FE_H (Access: Byte, Halfword, Word)**
- **VRAM19 (SEG38, SEG39): Address 0FF_H (Access: Byte, Halfword, Word)**

7	6	5	4	3	2	1	0	bit
D07	D06	D05	D04	D03	D02	D01	D00	
X	X	X	X	X	X	X	X	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

Regardless of the operation of LCD Controller/V driver, RAM can be read and written any time.

Figure 47.4-1 Correspondence between VRAM and Common/Segment Pins

Address						
0x0EC	VRAM0	bit 3	bit 2	bit 1	bit 0	SEG0
		bit 7	bit 6	bit 5	bit 4	SEG1
0x0ED	VRAM1	bit 3	bit 2	bit 1	bit 0	SEG2
		bit 7	bit 6	bit 5	bit 4	SEG3
0x0EE	VRAM2	bit 3	bit 2	bit 1	bit 0	SEG4
		bit 7	bit 6	bit 5	bit 4	SEG5
0x0EF	VRAM3	bit 3	bit 2	bit 1	bit 0	SEG6
		bit 7	bit 6	bit 5	bit 4	SEG7
0x0F0	VRAM4	bit 3	bit 2	bit 1	bit 0	SEG8
		bit 7	bit 6	bit 5	bit 4	SEG9
0x0F1	VRAM5	bit 3	bit 2	bit 1	bit 0	SEG10
		bit 7	bit 6	bit 5	bit 4	SEG11
0x0F2	VRAM6	bit 3	bit 2	bit 1	bit 0	SEG12
		bit 7	bit 6	bit 5	bit 4	SEG13
0x0F3	VRAM7	bit 3	bit 2	bit 1	bit 0	SEG14
		bit 7	bit 6	bit 5	bit 4	SEG15
0x0F4	VRAM8	bit 3	bit 2	bit 1	bit 0	SEG16
		bit 7	bit 6	bit 5	bit 4	SEG17
0x0F5	VRAM9	bit 3	bit 2	bit 1	bit 0	SEG18
		bit 7	bit 6	bit 5	bit 4	SEG19
0x0F6	VRAM10	bit 3	bit 2	bit 1	bit 0	SEG20
		bit 7	bit 6	bit 5	bit 4	SEG21
0x0F7	VRAM11	bit 3	bit 2	bit 1	bit 0	SEG22
		bit 7	bit 6	bit 5	bit 4	SEG23
0x0F8	VRAM12	bit 3	bit 2	bit 1	bit 0	SEG24
		bit 7	bit 6	bit 5	bit 4	SEG25
0x0F9	VRAM13	bit 3	bit 2	bit 1	bit 0	SEG26
		bit 7	bit 6	bit 5	bit 4	SEG27
0x0FA	VRAM14	bit 3	bit 2	bit 1	bit 0	SEG28
		bit 7	bit 6	bit 5	bit 4	SEG29
0x0FB	VRAM15	bit 3	bit 2	bit 1	bit 0	SEG30
		bit 7	bit 6	bit 5	bit 4	SEG31
0x0FC	VRAM16	bit 3	bit 2	bit 1	bit 0	SEG32
		bit 7	bit 6	bit 5	bit 4	SEG33
0x0FD	VRAM17	bit 3	bit 2	bit 1	bit 0	SEG34
		bit 7	bit 6	bit 5	bit 4	SEG35
0x0FE	VRAM18	bit 3	bit 2	bit 1	bit 0	SEG36
		bit 7	bit 6	bit 5	bit 4	SEG37
0x0FF	VRAM19	bit 3	bit 2	bit 1	bit 0	SEG38
		bit 7	bit 6	bit 5	bit 4	SEG39
				←	→	RAM area and common pins used in 1/2 duty cycle output mode
			←		→	RAM area and common pins used in 1/3 duty cycle output mode
		←			→	RAM area and common pins used in 1/4 duty cycle output mode

MB91460 Series**47.4.4 LCR1: LCDC Control Register 1**

- **LCR1H: Address 0EA_H (Access: Byte, Half-word, Word)**

15	14	13	12	11	10	9	8	bit
-	-	-	-	-	-	SEGEN9	SEGEN8	
-	-	-	-	-	-	0	0	Initial value
-	-	-	-	-	-	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit15-10: Undefined (Read: Indeterminate, Write: “0” is always written.)
- bit9-8: Segment driver enable.

Always set to “11_B” when LCD is used.

- **LCR1L: Address 0EB_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit7-0: Segment driver enable.

When LCD is used, always set this register to “11111111_B”.

47.4.5 LCDCMR: Common Pin Switching Register

- **LCDCMR: Address 0E8_H (Access: Byte, Half-word, Word)**

7	6	5	4	3	2	1	0	bit
DTCH	—	FPERIOD	—	COMEN3	COMEN2	COMEN1	COMEN0	
0	—	0	—	0	0	0	0	Initial value
R/W	RX/WX	R/W	RX/WX	R/W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit7: Analogue macro control.
Always set to “0_B” when LCD is used.
- bit6,4: Undefined (Read: Indeterminate, Write: Always write “0” to these bits.)
- bit3-0: Common driver enable.
Always set to “1111_B” when LCD is used.
- bit 5: FPERIOD (MB91FV460B only)
Select alternative frame period settings

FPERIOD	Operation
0	The frame period settings are like listed in LCR0 register description
1	The frame periods are elongated by factor 2

47.5. Operation

This section describes operation.

47.5.1 LCD Controller/Driver (LCDC) Operation

- (1) Set values to the display data memory (VRAM) in advance.
- (2) Make necessary settings to each register.
- (3) When the frame period generation clock oscillates, LCD drive waveform is output through common/segment output pins (COM0 - COM3, SEG0 - SEG39).
- (4) In more detail:
VRAM contents are automatically read in synchronization with common signals to be output through segment output pins.
(If the bit is set to "1", the selected waveform is output through the segment output pins.
If the bit is set to "0", non-selected waveform is output through the segment output pins.)
If the display mode is set to 1/2 duty cycle, non-selected waveform is output through the COM2 and COM3 pins. For the 1/3 duty cycle, the COM3 pin is used to output non-selected waveform.
- (5) This output waveform is a 2-frame AC waveform in accordance with the duty cycle setting, and drives LCD.
- (6) When MS[1:0] = "00" is used to deactivate LCD, a "L" level is output through both common and segment pins.
- (7) If LCD operation is enabled in the Sub-STOP state (LCEN="1"), LCD display is displayed¹.
Note that frame period generation clock signals must be supplied at this time.
- (8) LCD display can be blanked by selecting "blank" (BK="1") in blanking selection.
Note that non-selected waveform continues to be output.
- (9) When LCD deactivation (MS[1:0]="00") is selected with the display mode, LCD ceases to operate.

47.5.2 1/2 Duty Cycle Output Waveform

Only COM0 and COM1 outputs are used for LCD display. Neither COM2 nor COM3 output is used.

- Example of 1/3 Bias Output Waveform

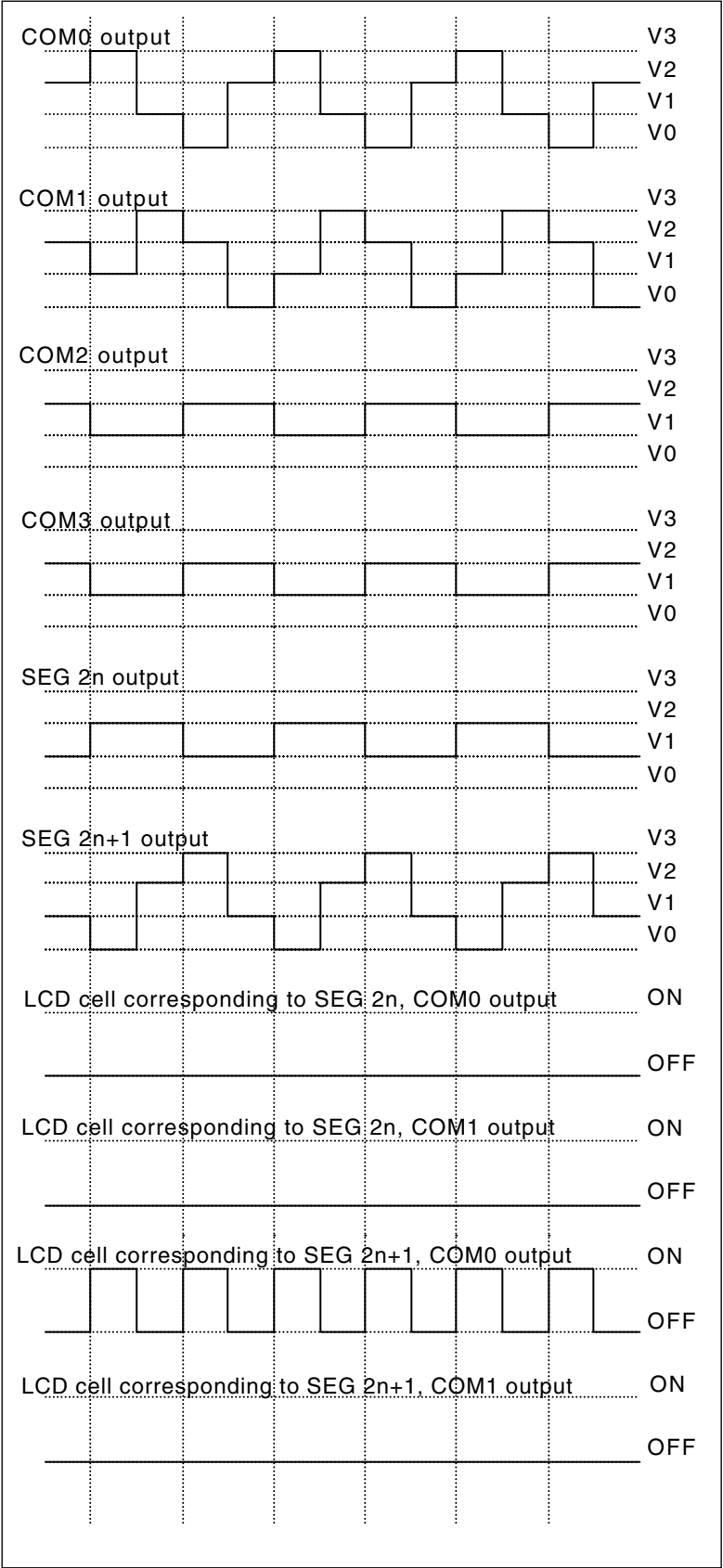
LCD cells with the maximum voltage difference between common and segment outputs are lit.

Table 47.5-1 Example of Data Memory Contents for display

Segment	Contents of data memory for display			
	COM3 output	COM2 output	COM1 output	COM0 output
SEG 2n output	-	-	0	0
SEG2n+1 output	-	-	0	1

1. LCD is displayed on MB91FV460B only.

Figure 47.5-1 1/2 Duty Cycle Output Waveforms



47.5.3 1/3 Duty Cycle Output Waveform

In the 1/3 duty cycle output mode, COM0, COM1 and COM2 outputs are used for LCD display. COM3 output is not used.

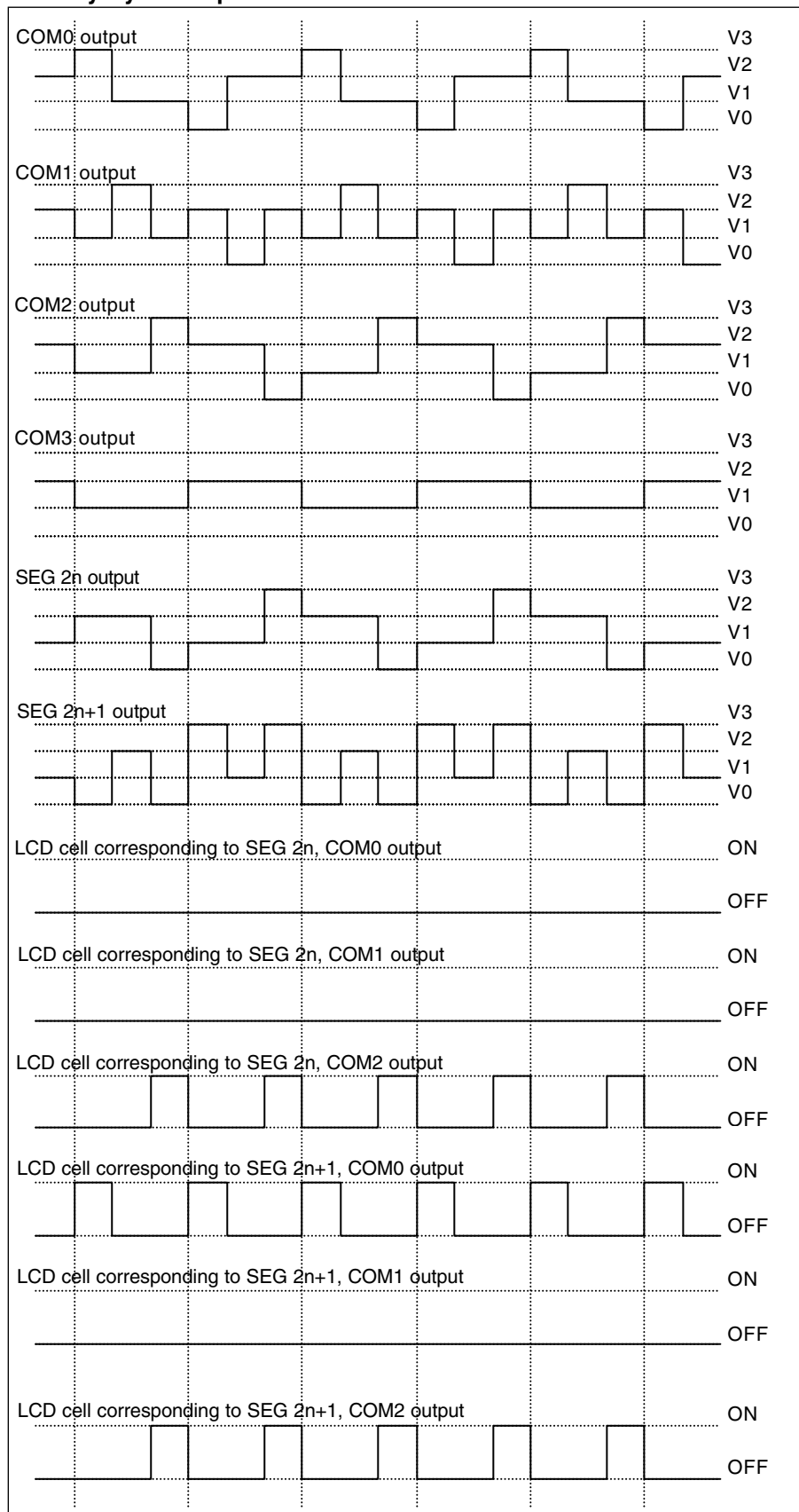
- Example of 1/3 Bias Output Waveform

LCD cells with the maximum voltage difference between common and segment outputs are lit.

Table 47.5-2 Example of Data Memory Contents for Display

Segment	Contents of data memory for display			
	COM3 output	COM2 output	COM1 output	COM0 output
SEG 2n output	-	1	0	0
SEG2n+1 output	-	1	0	1

Figure 47.5-2 1/3 Duty Cycle Output Waveforms



47.5.4 1/4 Duty Cycle Output Waveform

In the 1/4 duty cycle output mode, COM0, COM1, COM2, and COM3 outputs are all used for LCD display.

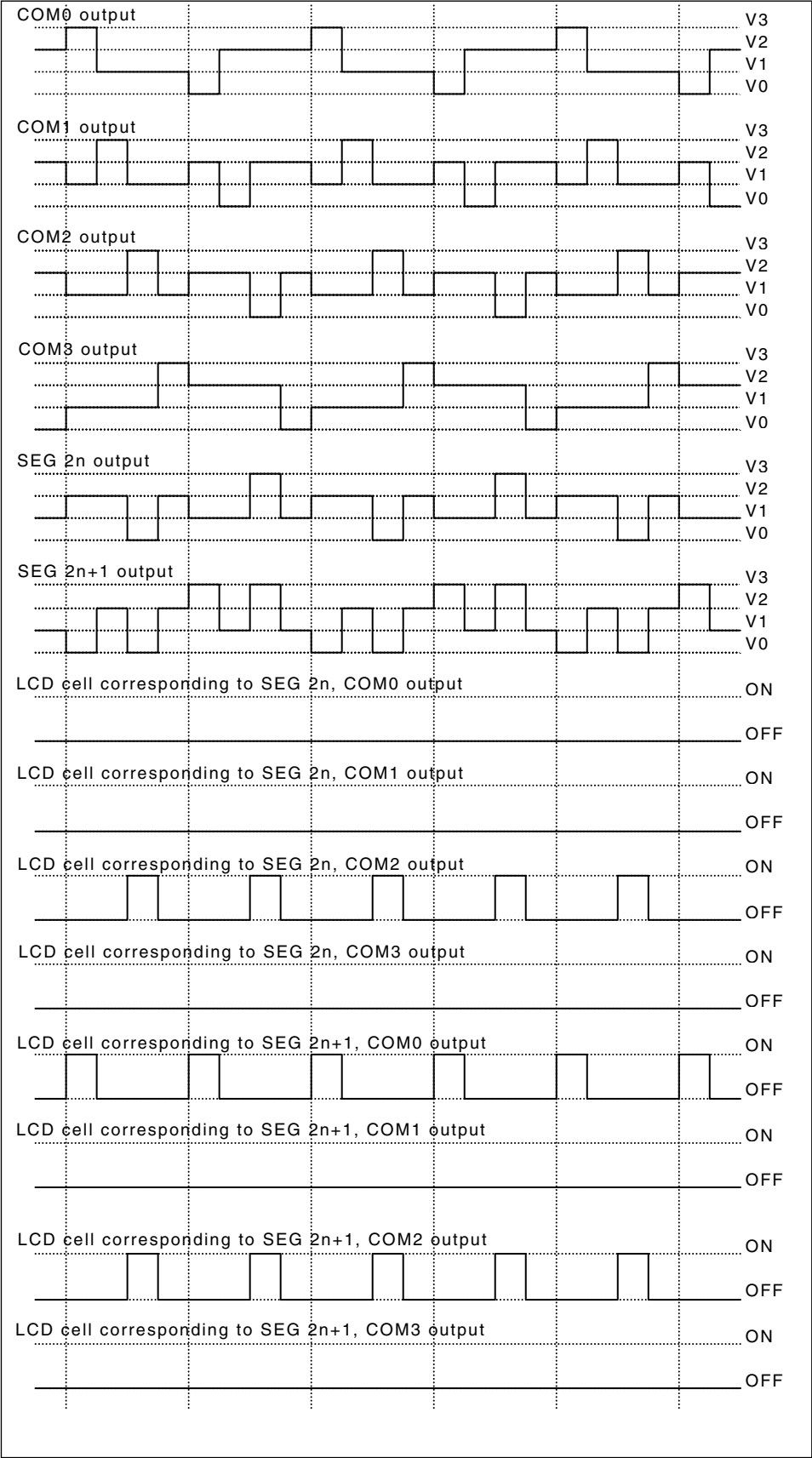
- Example of 1/3 Bias Output Waveform

LCD cells with the maximum voltage difference between common and segment output are lit.

Table 47.5-3 Example of Data Memory Contents for Display

Segment	Contents of data memory for display			
	COM3 output	COM2 output	COM1 output	COM0 output
SEG 2n output	0	1	0	0
SEG2n+1 output	0	1	0	1

Figure 47.5-3 1/4 Duty Cycle Output Waveforms



47.6. Setting

Table 47.6-1 Required Setting to Use LCD

Setting	Setting register *	Setting procedure
Presetting	Common pin switching register (LCDCMR) LCD control register 1 (LCR1)	–
Set divided resistors.	LCD control register 0 (LCR0)	See 7.8 and 7.9
Set ports	Port function register (PFR)	See 7.1
Set display data.	Display data memory (VRAM)	See 7.2
Select the frame period generation clock. Set a frame period.	LCD control register 0 (LCR0)	See 7.3
Select a duty cycle. (Activation)		See 7.4
Enable LCD display.		See 7.6

* :For the setting procedure, refer to the section indicated by the number.

Table 47.6-2 Required Setting to Disable LCD display

Setting	Setting register *	Setting procedure
Disable (blank) LCD display.	LCD control register 0 (LCR0)	See 7.6

* :For the setting procedure, refer to the section indicated by the number.

Table 47.6-3 Required Setting to Deactivate LCD

Setting	Setting register *	Setting procedure
Deactivate LCD.	LCD control register 0 (LCR0)	See 7.5

*:For the setting procedure, refer to the section indicated by the number.

Table 47.6-4 Required Setting to Enable LCD Display in Sub-STOP state¹

Setting	Setting register *	Setting procedure
Enable LCD display in the Sub-STOP state.	LCD control register 0 (LCR0)	See 7.7
Select the frame period generation clock.		See 7.3
Switch to Sub clock operation.	Set LCR0.CSS ^{*2}	–
Change to the STOP state.	See “ Chapter 10 Standby (Page No.241) ”.	–

1. Possible on MB91FV460B only.

2. The CPU does not need to be switched to Sub Clock. It is sufficient to switch the LCD clock source to Sub Clock by setting LCR0.CSS

* :For the setting procedure, refer to the section indicated by the number.

MB91460 Series**47.7. Q&A****47.7.1 How to set pins as COM or SEG output pins**

Use COM and SEG output settings.

Software can switch ports to COM or SEG output ports.

To do so, write “1” to the output designation bit (COM[3:0], SEG[39:0]).

Pin	Register	Output designation bit
COM0	Port function register PFR30[3:0]	(COM0)
COM1		(COM1)
COM2		(COM2)
COM3		(COM3)
SEG0	Port function register PFR35[7:0]	(SEG0)
SEG1		(SEG1)
SEG2		(SEG2)
SEG3		(SEG3)
SEG4		(SEG4)
SEG5		(SEG5)
SEG6		(SEG6)
SEG7		(SEG7)
SEG8	Port function register PFR34[7:0]	(SEG8)
SEG9		(SEG9)
SEG10		(SEG10)
SEG11		(SEG11)
SEG12		(SEG12)
SEG13		(SEG13)
SEG14		(SEG14)
SEG15		(SEG15)
SEG16	Port function register PFR33[7:0]	(SEG16)
SEG17		(SEG17)
SEG18		(SEG18)
SEG19		(SEG19)
SEG20		(SEG20)
SEG21		(SEG21)
SEG22		(SEG22)
SEG23		(SEG23)
SEG24	Port function register PFR32[7:0]	(SEG24)
SEG25		(SEG25)
SEG26		(SEG26)
SEG27		(SEG27)
SEG28		(SEG28)
SEG29		(SEG29)
SEG30		(SEG30)
SEG31		(SEG31)

SEG32	Port function register PFR31[7:0]	(SEG32)
SEG33		(SEG33)
SEG34		(SEG34)
SEG35		(SEG35)
SEG36		(SEG36)
SEG37		(SEG37)
SEG38		(SEG38)
SEG39		(SEG39)

MB91460 Series

47.7.2 How to set VRM

The following tables show the relationship between pins and the bit positions of VRAM(n). (n=0 to 19)

Table 47.7-1 1/2 duty cycle

Pin	COM1	COM0
SEG 2n	bit 1	bit 0
SEG 2n+1	bit 5	bit 4

Table 47.7-2 1/3 duty cycle

pin	COM2	COM1	COM0
SEG 2n	bit 2	bit 1	bit 0
SEG 2n+1	bit 6	bit 5	bit 4

Table 47.7-3 1/4 duty cycle

pin	COM3	COM2	COM1	COM0
SEG 2n	bit 3	bit 2	bit 1	bit 0
SEG 2n+1	bit 7	bit 6	bit 5	bit 4

(Non-selected waveform is output through the pins other than the above.)

Example: 1/4 duty cycle

When “1” is set to the bit6 of VRAMn, selected waveform is output through the SEG2n+1 of COM2.

If a bit is set to “0”, non-selected waveform is output through the corresponding pin.

47.7.3 How to set a frame period

Use the frame period generation clock select bit (LCR0.CSS) and the frame period bit (LCR0.FP[1:0]). The following settings are available:

Frame period (MB91FV460B)	Selected value	
	Frame period generation clock select bit (CSS)	Frame period bit (FP[1:0])
$CLKP/(2^{12} \times N)$	Set to “0”.	Set to “00”.
$CLKP/(2^{14} \times N)$	Set to “0”.	Set to “01”.
$CLKP/(2^{16} \times N)$	Set to “0”.	Set to “10”.
$CLKP/(2^{18} \times N)$	Set to “0”.	Set to “11”.
$CLKSUB/(2^7 \times N)$	Set to “1”.	Set to “00”.
$CLKSUB/(2^8 \times N)$	Set to “1”.	Set to “01”.
$CLKSUB/(2^9 \times N)$	Set to “1”.	Set to “10”.
$CLKSUB/(2^{10} \times N)$	Set to “1”.	Set to “11”.

N (Time division number) = MS[1:0] value + “1”

Set an appropriate frame period that corresponds to the frame frequency of the LCD panel.

47.7.4 How to set a duty cycle

Use the display mode select bit (LCR0.MS[1:0]).

Controlled operation	Display mode select bit (MS[1:0])	N (Time division number)
To deactivate LCD (Pin output: "L")	Set to "00".	N/A
To set the 1/2 duty cycle output mode	Set to "01".	2
To set the 1/3 duty cycle output mode	Set to "10".	3
To set the 1/4 duty cycle output mode	Set to "11".	4

The display mode select bit also serves as an operation start/stop control bit.

47.7.5 How to control starting and stopping of LCD

Use the display mode select bit (LCR0. MS[1:0]) to control start and stop of operation.

See (4).

47.7.6 How to enable or disable LCD display

Use either of the following methods:

- Use the blanking select bit (LCR0. BK).

Controlled operation	Blanking select bit (BK)
To enable LCD display	Set to "0".
To disable (blank) LCD display (Non-selected waveform is output through segment pins.)	Set to "1".

- The LCD display can be blanked by using the display mode select bit (LCR0. MS[1:0]) to select LCD deactivation.

Controlled operation	Display mode select bit (MS[1:0])
To deactivate LCD (A "L" level is output through common and segment pins.)	Set to "00".

47.7.7 How to enable LCD display even in the Sub-STOP state

Use the Sub-STOP operation enable bit (LCR0. LCEN).

Controlled operation	Frame period generation clock select bit (CSS)	Sub-STOP operation enable bit (LCEN) ¹
To disable LCD display during Sub-STOP	—	Set to "0".
To enable LCD display during Sub-STOP	-	
To enable LCD display when the Main clock stops and the Sub clock operates	Set to "1".	—

1. On MB91V460A it is not possible to enable the LCD output in any kind of STOP mode. The analog part of LCD is always disabled in STOP mode here. The LCD of MB91FV460B can work in STOP mode.

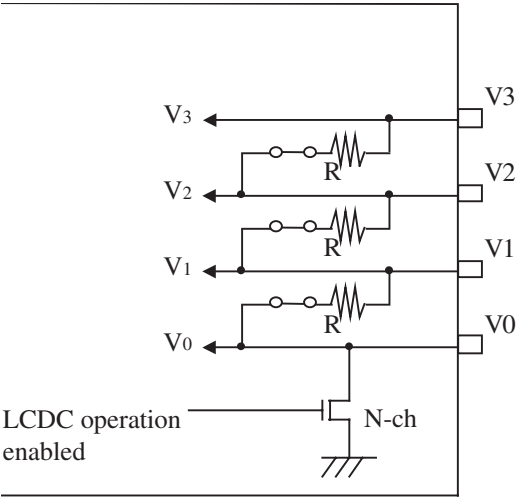
47.7.8 How to select internal or external divided resistors

Use the LCD drive power supply control bit (LCR0.VSEL).

Controlled operation	LCD drive power supply control bit (VSEL).
To use external divided resistors (Internal divided resistors disconnected)	Set to "0".
To use internal divided resistors (Internal divided resistors connected)	Set to "1".

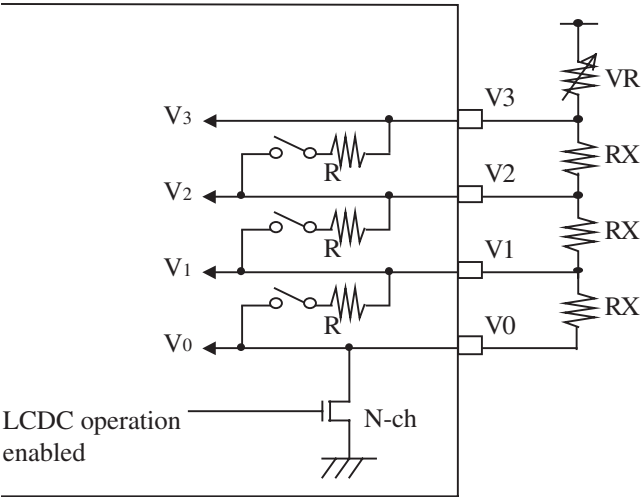
47.7.9 How to select internal or external divided resistors

Figure 47.7-1 Selection of internal divided resistors



- When using external divided resistors:
The LCD driving voltage can be generated by connecting external divided resistors to the LCD drive power supply pins (V0 to V3).

Figure 47.7-2 Selection of external divided resistors



To avoid the effect of internal divided resistors, the resistors must be disconnected by setting "0" to the LCD drive power supply control bit (LCR0.VSEL).

47.7.10 How to use external divided resistors to shut off the current when LCD is deactivated

The V0 pin is internally connected to Vss (GND) via a transistor. For this reason, the current generated on deactivating LCD controller can be shut off by connecting external divided resistors to the V0 pin on the Vss side. To shut off the current, use the display mode select bit (MS[1:0]= "00").

47.8. Caution

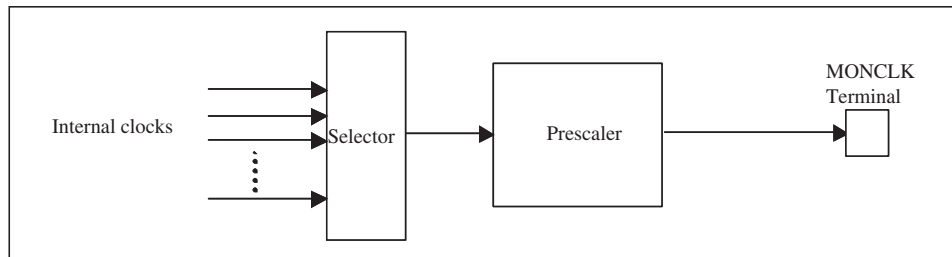
- VRAM[0:19] can be written in byte, halfword and word access.
- Switching the frame period generation clocks:
Frame period generation clocks (LCR0:CSS) can be switched even during LCD display. However, switching may cause some screen flicker. To avoid such flicker, be sure to set the blanking select bit (LCR0:BK) to "1" (blank display) before switching.
- Depending on the LCD, different external divided resistors are used. Use appropriate resistor values.
- When the display mode is set to 1/2 duty cycle, non-selected waveform is output through the COM2 and COM3 pins. For 1/3 duty cycle, the COM3 pin is used to output non-selected waveform.
- Inappropriate selection or setting of frame period generation clock (CSS), LCD drive power supply control (VSEL), duty cycle (MS[1:0]) and frame period (FP[1:0]) results in inappropriate LCD display.
- MB91V460A: The LCD is disabled in any kind of STOP state.(See 6 "Setting procedure".)

Chapter 48 Clock Monitor

48.1. Overview

The Clock Monitor is a module that outputs internal clock signals to a terminal to externally monitor them. The Clock Monitor provides a function to divide the frequency of a clock signal before it outputs to the terminal, thus allowing the clock signal to be used as an event at which external circuits act in synchronization with a MCU function.

Figure 48.1-1 Block diagram of Clock Monitor



48.2. Features

- Format: Divide an internal clock signal to output it to a terminal (MONCLK).
- Channel: 1
- Division ratios: CLK/1, CLK/2, CLK/3, ..., CLK/16
- Glitch free output enable
- Programmable mark level (output “L” or “H” before enabling the clock output)
- Interrupt: None

48.3. Configuration

Figure 48.3-1 Configuration Diagram of Clock Monitor

Clock Monitor

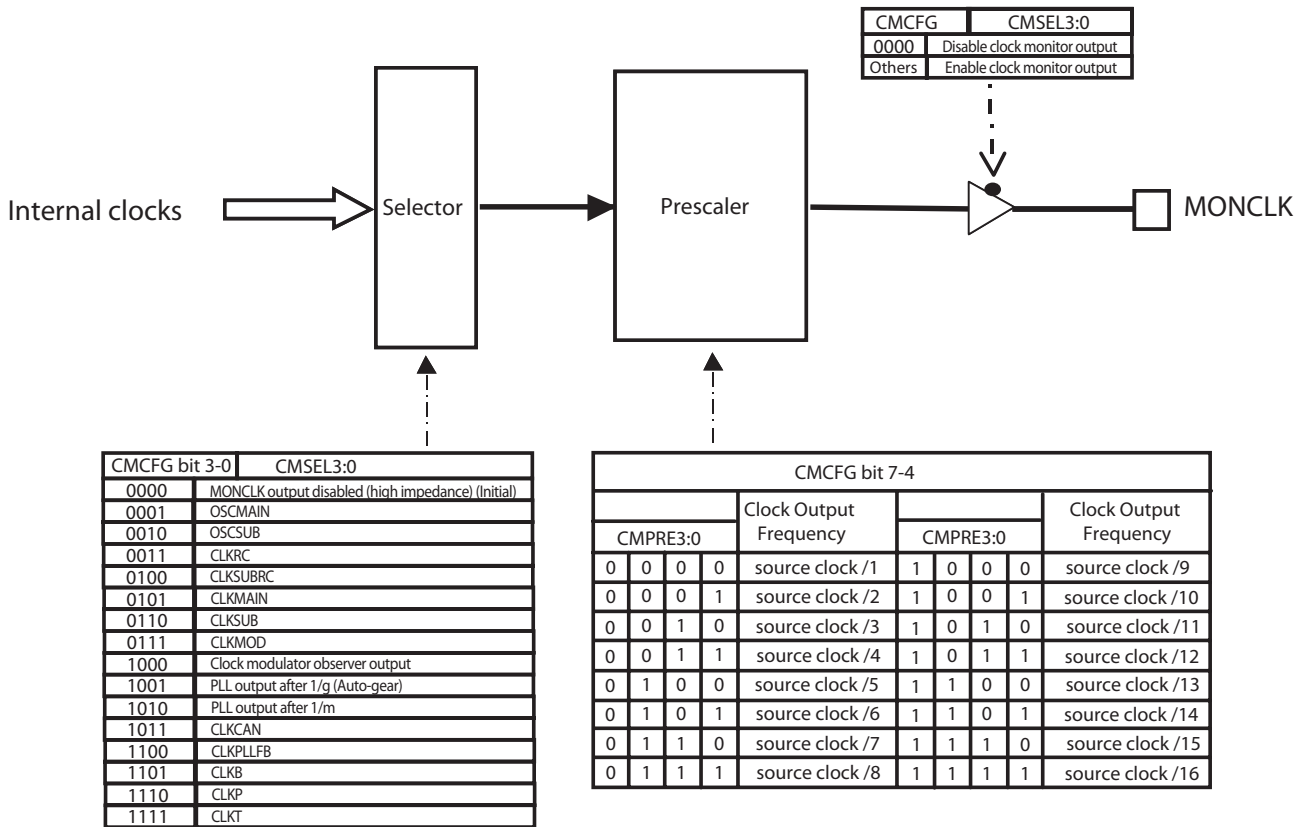


Figure 48.3-2 Register List

Clock Monitor

Address	Bit	7	6	5	4	3	2	1	0		
0004AEH		EDSUE	---	---	MONCKI	CSC3	CSC2	CSC1	CSC0	CSCFG	(Clock source config)
0004AFH		CMPRE3	CMPRE2	CMPRE1	CMPRE0	CMSEL3	CMSEL2	CMSEL1	CMSEL0	CMCFG	(Clock monitor config)

MB91460 Series

48.4. Register

48.4.1 Clock Monitor Configuration Register

A register for output settings of an internal clock signal.

- **CMCFG: Address 04AF_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
CMPRE3	CMPRE2	CMPRE1	CMPRE0	CMSEL3	CMSEL2	CMSEL1	CMSEL0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attributes

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit7-4: Select an output frequency prescaler

CMPRE3	CMPRE2	CMPRE1	CMPRE0	Clock Frequency Output to the MONCLK pin
0	0	0	0	Source clock (selected by CMSEL) divided by 1 (Initial)
0	0	0	1	Source clock (selected by CMSEL) divided by 2
0	0	1	0	Source clock (selected by CMSEL) divided by 3
0	0	1	1	Source clock (selected by CMSEL) divided by 4
0	1	0	0	Source clock (selected by CMSEL) divided by 5
0	1	0	1	Source clock (selected by CMSEL) divided by 6
0	1	1	0	Source clock (selected by CMSEL) divided by 7
0	1	1	1	Source clock (selected by CMSEL) divided by 8
1	0	0	0	Source clock (selected by CMSEL) divided by 9
1	0	0	1	Source clock (selected by CMSEL) divided by 10
1	0	1	0	Source clock (selected by CMSEL) divided by 11
1	0	1	1	Source clock (selected by CMSEL) divided by 12
1	1	0	0	Source clock (selected by CMSEL) divided by 13
1	1	0	1	Source clock (selected by CMSEL) divided by 14
1	1	1	0	Source clock (selected by CMSEL) divided by 15
1	1	1	1	Source clock (selected by CMSEL) divided by 16

- bit3-0: Select a clock source

CMSEL	CMSEL2	CMSEL1	CMSEL0	Clock Source Output to the MONCLK pin
0	0	0	0	MONCLK output disabled (high impedance) (Initial)
0	0	0	1	OSCMAN (Main oscillation before Clock Supervisor)
0	0	1	0	OSCSUB (Sub oscillation before Clock Supervisor)
0	0	1	1	CLKRC (RC Clock, either 100kHz or 2MHz)
0	1	0	0	CLKSUBRC (Sub clock for Base Clock generation) ¹
0	1	0	1	CLKMAIN (Main oscillation after Clock Supervisor)
0	1	1	0	CLKSUB (Sub oscillation after Clock Supervisor)
0	1	1	1	CLKMOD (Output from Clock Modulator)
1	0	0	0	Clock modulator observer output ²
1	0	0	1	PLL output after 1/g divider (Auto Gear)
1	0	1	0	PLL output after 1/m divider
1	0	1	1	CLKCAN (CAN clock)
1	1	0	0	CLKPLLFB (PLL feedback)
1	1	0	1	CLKB

1	1	1	0	CLKP
1	1	1	1	CLKT

1. If Media LB module is installed and the Media LB PLL is enabled, Media LB clocks are output.
2. For silicon evaluation purposes only, not available for customer applications

48.4.2 Clock Source Configuration Register

For the Clock Monitor, this register contains the MONCLKI bit to control the MONCLK mark level.

- **CSCFG: Address 04AEh (Access: Byte)**

7	6	5	4	3	2	1	0	bit
EDSUEN	PLLLOCK	RCSEL	MONCKI	CSC3	CSC2	CSC1	CSC0	
0	X	0	0	0	0	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (software reset)
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- bit4: Clock Monitor MONCLK inverter

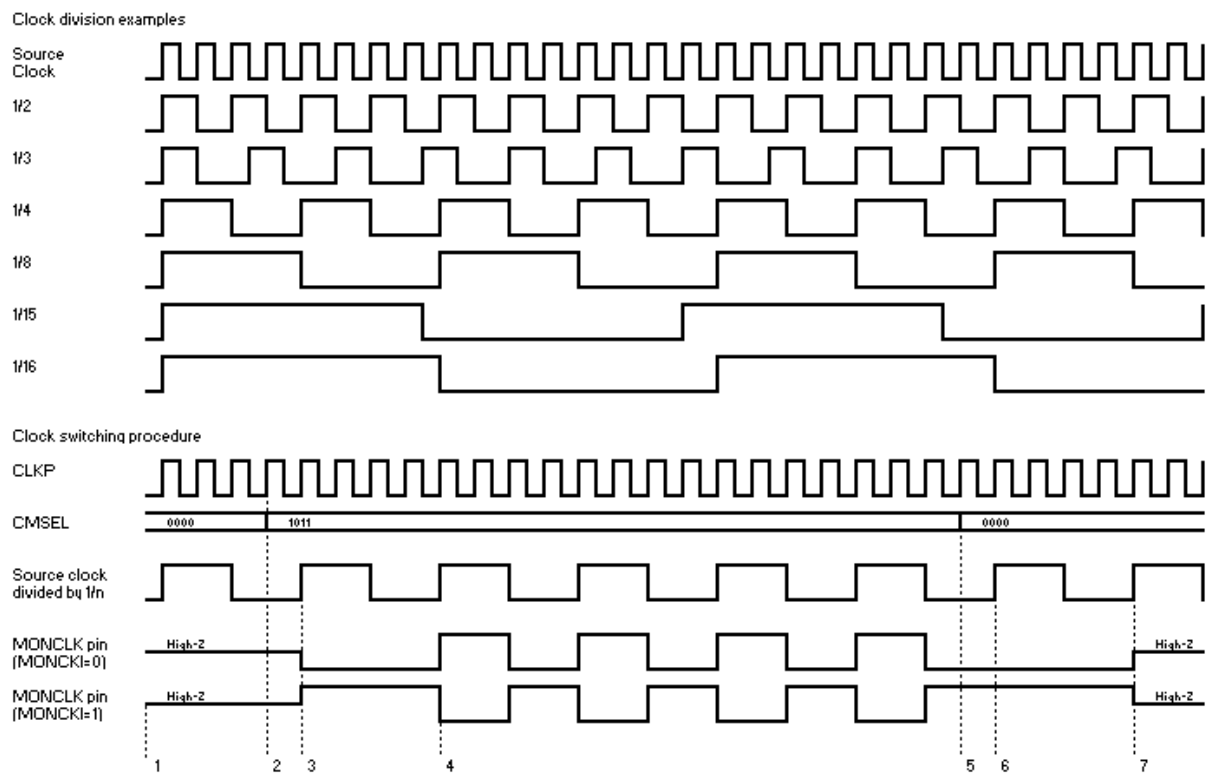
MONCKI	Function
0	MONCLK mark level is low [Initial value]
1	MONCLK mark level is high

- Other bits7-5, bits3-0

Not belonging to clock monitor operation. See chapter “[13.4.5 CSCFG: Clock Source Configuration Register \(Page No.299\)](#)” about information on the additional functions of this register.

48.5. Operation

Figure 48.5-1 Output waveforms of the Clock Monitor



- (1) The MONCLK pin is in high impedance state.
- (2) CMSEL is set from "0000" (no clock selected) to the selected (and prescaled) clock.
- (3) The MONCLK pin changes to output "L" status (output "H" if MONCKI is set to '1') for one period of the internal (prescaled) clock.
- (4) After one period of the selected (and prescaled) internal clock MONCLK outputs the selected (and prescaled) internal clock.
- (5) CMSEL is set from the selected clock to "0000" (no clock selected).
- (6) The MONCLK pin changes to output "L" status (output "H" if MONCKI is set to '1') for one period of the internal (prescaled) clock.
- (7) The MONCLK pin switches to high impedance state.

48.6. Settings

Table 48.6-1 Settings for Using Clock Monitor

Settings	Setting Registers	Setting Procedure*
Set a prescaler value	Clock Monitor Prescaler (CMCFG.CMPRE[3:0])	See 48.7.2
Set a source clock	Clock Monitor Selection (CMCFG.CMSEL[3:0])	See 48.7.1
Change the mark level	Clock Monitor Inverter (CSCFG.MONCKI)	See 48.4.2
Enable clock monitor output.(MONCLK)	Clock Monitor Selection (CMCFG.CMSEL[3:0])	See 48.7.1
		See 48.7.3

*:For each setting procedure, refer to an appropriate section.

48.7. Q&A

48.7.1 How to set an output terminal (MONCLK)

Use the Clock Monitor Selection bits (CMCFG.CMSEL[3:0])

Operation	CMCFG.CMSEL[3:0]
To set an output terminal (MONCLK)	Set to the appropriate clock (!= "0000")

48.7.2 How to select an output frequency

Use the Output Frequency Select bit (CMCFG.CMPRE[3:0]).

Clock Division Ratio	Output Frequency (Example)		Frequency Prescaler (CMCFG.CMPRE[3:0])
	CLKP=32MHz	CLKP=40MHz	
1/2	16.0MHz	20.0MHz	Set to "0001".
1/3	10.7MHz	13.3MHz	Set to "0010".
1/4	8.0MHz	10.0MHz	Set to "0011".
1/8	4.0MHz	5.0MHz	Set to "0111".
1/15	2.1MHz	2.7MHz	Set to "1110".
1/16	2.0MHz	2.5MHz	Set to "1111".

48.7.3 How to enable/disable clock monitor output

Use the Output Enable bit (CMCFG.CMSEL[3:0]).

Operation	Output Enable Bit (CMCFG.CMSEL[3:0])
To disable clock monitor output (To set the terminal to the Hi-z status)	Set to "0000".
Enable clock monitor output.	Set to "0001"-"1111".

MB91460 Series

48.8. Caution

Due to the glitch free switching mechanism it is necessary to follow these rules when switching the clock source (CMCFG[3:0]) or the prescaler ratio (CMPRE[3:0]):

- The CMPRE[3:0] registers can only be written if the CMSEL[3:0] registers are currently 0x0.
- The CMPRE[3:0] registers can only be written if the CMSEL[3:0] registers are written to 0x0 within the same write access.
- Between 2 write accesses to CMPRE/CMCFG there must be at least 2 cycles of the divided monitor clock.
- For changing the selector value, the MONCLK must be disabled.

Example access:

1. access:

CMCFG_CMSEL = 0

CMCFG_CMPRE = prescaler

2. access:

CMCFG_CMSEL = clock

Chapter 49 Real-Time Clock (RTC)

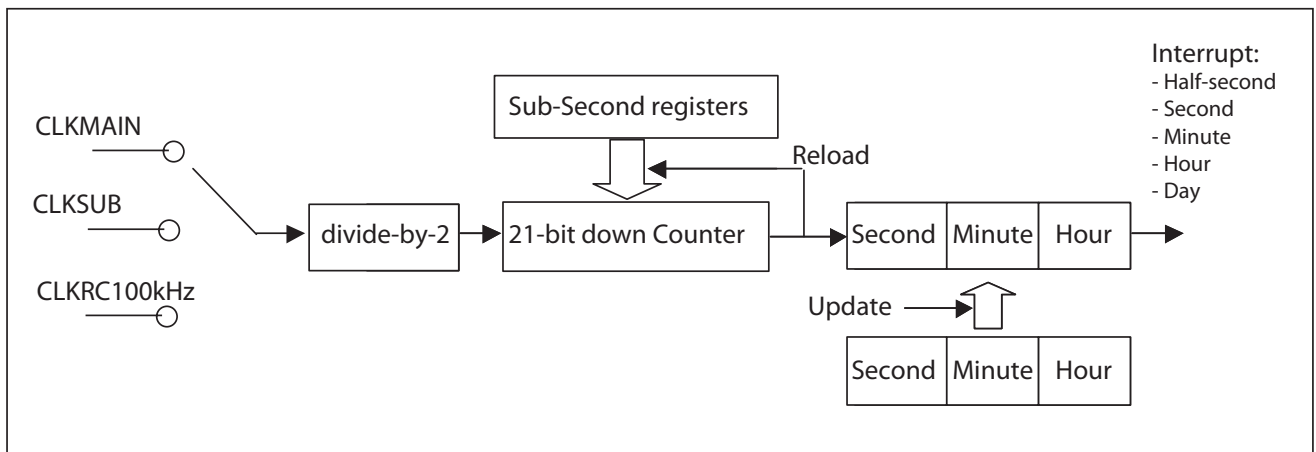
49.1. Overview

Real-time Clock (RTC) continues to count elapsed time even in the STOP state to provide the current real time (HH/MM/SS) based on CLKMAIN (4MHz), CLKSUB (32kHz) or CLKRC100kHz (~100kHz).

The CLKSUB and the CLKRC 100kHz can be calibrated with respect to the CLKMAIN. (See [Chapter 50 Clock Calibration Unit \(Page No.1145\)](#))

This allows precise time counting without a return from an interrupt during standby periods.

Figure 49.1-1 Block diagram of the Real-Time Clock



49.2. Features

- Information: Time count (HH/MM/SS). (This clock continues to operate even in the STOP state.)
- Operational on CLKMAIN (4MHz), CLKSUB (32kHz) or CLKRC (~100kHz)
(The operational clock is selected with the bits CSCFG.CSC1-0. See [“Bit1-0: Clock Source Selection for RTC” on page 300.](#))
- Quantity: 1
- Time unit: selected clock divided by 2
- Operation clock:
 - For register access: CLKP
 - For time count: CLKMAIN, CLKSUB, CLKRC100kHz
- Time: Initial setting and adjustment are possible.
- Interrupt: Interrupts can be generated at any of the five intervals: 1 half-second, 1 second, 1 minute, 1 hour and 1 day.
- Others: By changing the value of the sub-second register, interrupts can be generated at any interval (from short to long).

49.3. Configuration

Figure 49.3-1 Configuration of the Real-Time Clock

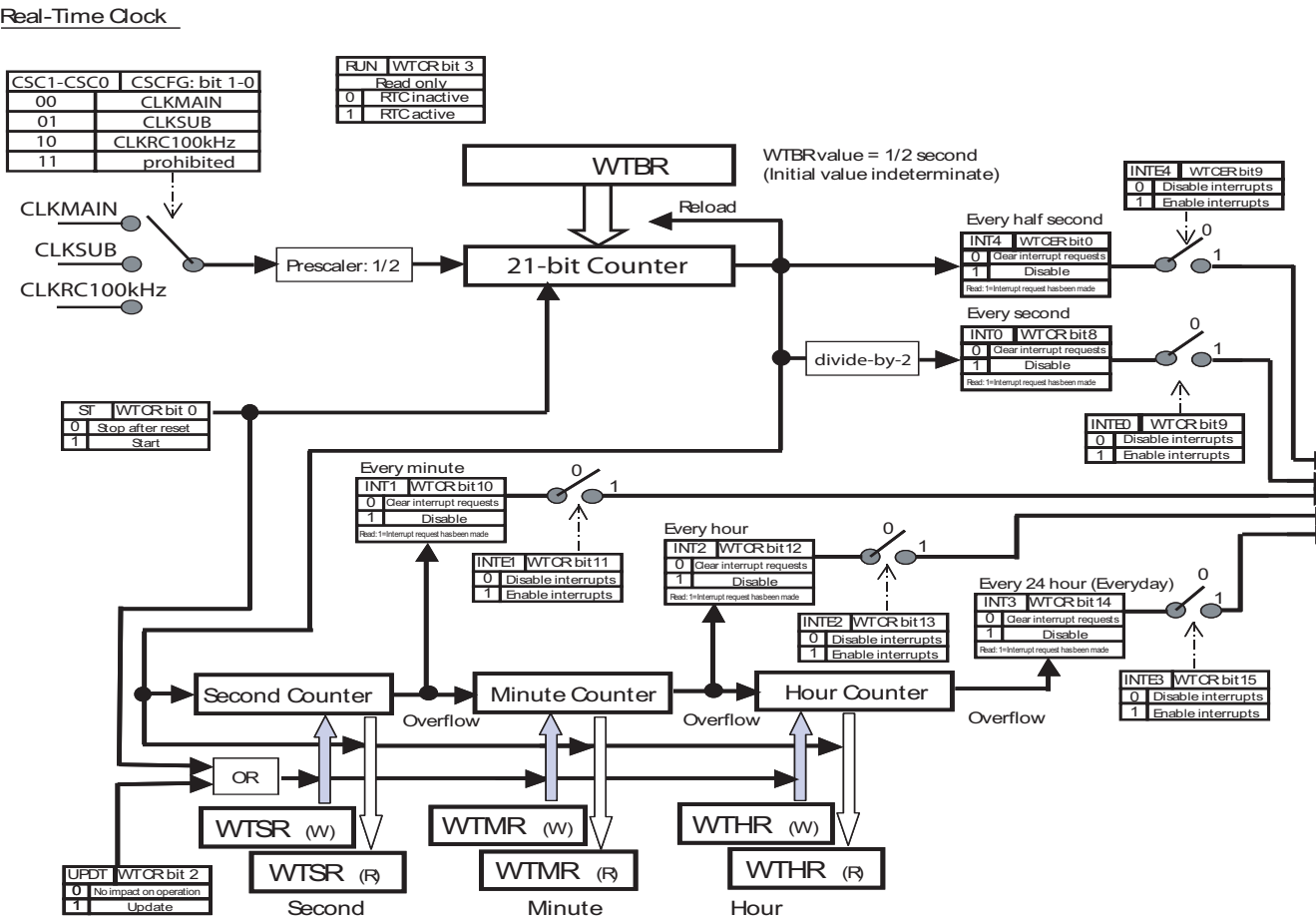


Figure 49.3-2 Register List of the Real-Time Clock

Real-Time Clock

Address	Bit	7	6	5	4	3	2	1	0									
0004A1H		---	---	---	---	---	---	INT4	INT4	WTCER								
0004A2H	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		INT3	INT3	INT2	INT2	INT1	INT1	INT0	INT0	TST2	TST1	TST0	---	RUN	UPDT	---	ST	WTCR (RTC control)
0004A5H	Bit	7	6	5	4	3	2	1	0									
		---	---	---	D20	D19	D18	D17	D16									WTCR 0
0004A6H																		WTCR 1
0004A7H																		WTCR 2
0004A8H					H4	H3	H2	H1	H0									WTHR (Hour)
0004A9H				M5	M4	M3	M2	M1	M0									WTMR (Minute)
0004AAH				S5	S4	S3	S2	S1	S0									WTSR (Second)
0004AEH		EDS	UEN	PLL	LOCK	RCSEL	MONCK1	CSC3	CSC2	CSC1	CSC0							CSCFG* (Clock source configuration)
00047AH					ICR4	ICR3	ICR2	ICR1	ICR0									ICR58 (Interrupt level for RTC)
0FFDECH		32Bits																(Interrupt vector #132)

* For ICR registers and interrupt vectors, refer to the "Interrupt Control" section.
* For description of CSCFG register look at chapter 13 or in section 7 (Q & A) in this chapter

Note: For ICR registers and interrupt vectors, refer to "Chapter 24 Interrupt Control (Page No.429)".

MB91460 Series**49.4. Registers****49.4.1 WTCR: RTC Control Register**

This register is used to control behavior of the Real-time Clock module.

- **WTCR: Address 04A2_H (Access: Byte, Half-word)**

15	14	13	12	11	10	9	8	bit
INTE3	INT3	INTE2	INT2	INTE1	INT1	INTE0	INT0	
0	0	0	0	0	0	0	0	Initial value
0	0	0	0	0	0	0	0	When reset
R/W	R(R1),W	R/W	R(R1),W	R/W	R(R1),W	R/W	R(R1),W	Attribute

7	6	5	4	3	2	1	0	bit
Reserve d	Reserve d	Reserve d	–	RUN	UPDT	–	ST	
0	0	0	–	0	0	–	X	Initial value
0	0	0	–	0	0	–	X	When reset
R/W0	R/W0	R/W0	RX/WX	R/WX	R(R0)/W	RX/WX	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

- bit15: Enable interrupt requests at 1-day intervals

INTE3	Operation
0	Interrupt requests at 1-day (24 hour) interval disabled.
1	Interrupt requests at 1-day (24 hour) interval enabled.

- bit14: 1-day interrupt flag

INT3	Status	
	Read	Write
0	No interrupt is generated at 1-day (24 hour) interval.	Clear the flag.
1	An interrupt is generated at 1-day (24 hour) interval.	Writing does not affect the operation.

When the hour counter overflows, this flag is set to “1”.

- bit13: Enable interrupt requests at 1-hour intervals

INTE2	Operation
0	Interrupt request at 1-hour interval disabled.
1	Interrupt request at 1-hour interval enabled.

- bit12: 1-hour interrupt flag

INT2	Status	
	Read	Write
0	No interrupt is generated at 1-hour interval.	Clear the flag.
1	An interrupt is generated at 1-hour interval.	Writing does not affect the operation.

When the minute counter overflows, this flag is set to “1”.

- bit11: Enable interrupt requests at 1-minute intervals

INTE1	Operation
0	Interrupt request at 1-minute interval disabled.
1	Interrupt request at 1-minute interval enabled.

- bit10: 1-minute interrupt flag

INT1	Operation	
	Read	Write
0	No interrupt is generated at 1-minute interval.	Clear the flag.
1	An interrupt is generated at 1-minute interval.	Writing does not affect the operation.

When the second counter overflows, this flag is set to “1”.

- bit9: Enable interrupt requests at 1-second intervals

INTE0	Operation
0	Interrupt request at 1-second interval disabled.
1	Interrupt request at 1-second interval enabled.

- bit8: 1-second interrupt flag

INT0	Status	
	Read	Write
0	No interrupt is generated at 1-second interval.	Clear the flag.
1	An interrupt is generated at 1-second interval.	Writing does not affect the operation.

Every 2nd time when the 21-bit down counter is set to “0”, this flag is set to “1”.

- bit7-5: Reserved

Be sure to write “0”. The read value is the value written.

- bit4: Undefined

Writing does not affect the operation. The read value is indeterminate.

- bit3: Operation status

RUN	Status
0	The Real-time Clock module is inactive.
1	The Real-time Clock module is active.

- bit2: Update

UPDT	Status/Operation
0	The update has been completed. (Writing “0” does not affect the operation.)
1	Update the hour/minute/second counters with the values of the hour/minute/second registers, respectively.

Before writing “1” to the update bit (UPDT), the hour/minute/second registers must be set to the values with which to update the hour/minute/second counters. The hour/minute/second registers are updated on reloading to the 21-bit down counter.

- bit1: Undefined

Writing does not affect the operation. The read value is indeterminate.

- bit0: Start

ST	Operation
0	The Real-time Clock module stops to operate, and the 21-bit down counter and the hour/minute/second counters are cleared.
1	The settings of the hour/minute/second registers are loaded to the hour/minute/second counters, and the Real-time Clock module starts to operate.

Application Note: The Sub-second register of the RTC module stores the reload value for the 21-bit prescaler. This value is reloaded after the reload counter reaches "0". When modifying all three bytes, make sure the reload operation will not be performed in between the write instructions. Otherwise the 21-bit prescaler loads the incorrect value of the combination of new data and old data bytes. It is generally recommended that the Sub-Second register is updated while the ST bit is "0".

However, if this update is done immediately after an RTC second interrupt there should be enough time to securely modify the registers until the next reload operation (next second interrupt) even if ST is not set to "0" and the module is in operation.

When updating the registers by using the ST bit the following must be taken into account:

The new value is written into the registers with the rising edge of the RUN bit. This RUN bit is clocked by the RTC clock (32kHz, 100kHz or 4MHz depending on device and mode). To make sure that the update is done properly, write the new values into the registers, set ST to 0, wait for the RUN bit to go low and then start the circuit again by setting ST to 1. RUN will go low at the second rising edge of the RTC clock after ST has been set to 0. It will rise again at the half second rising edge of RTC clock after ST has been set to 1. If this operation is to be done several times directly after each other, wait for RUN to go to high before setting ST to low again.

- **WTCER: Address 04A1_H (Access: Byte)**

7	6	5	4	3	2	1	0	bit
—	—	—	—	—	—	INTE4	INT4	
—	—	—	—	—	—	0	0	Initial value
—	—	—	—	—	—	0	0	When reset
RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	RX/WX	R/W	R(R1),W	Attribute

(For attributes, refer to "[Meaning of Bit Attribute Symbols \(Page No.15\)](#)".)

- bit7-2: Undefined

Writing does not affect the operation. The read value is indeterminate.

- bit1: Enable interrupt requests at half-second (500ms) intervals

INTE4	Operation
0	Interrupt request at half-second (500 ms) interval is disabled.
1	Interrupt request at half-second (500 ms) interval is enabled.

- bit0: half-second (500ms) interrupt flag

INT4	Status	
	Read	Write
0	No interrupt is generated at half-second (500 ms) interval.	Clear the flag.
1	An interrupt is generated at half-second (500 ms) interval.	Writing does not affect the operation.

When the 21-bit down counter is set to "0", this flag is set to "1".

49.4.2 WTBR: Sub-Second Registers

These registers are used to hold values to be reloaded to the 21-bit down counter.

- **WTBR0: Address 04A5_H (Access: Byte, Half-word, Word)**
- **WTBR1: Address 04A6_H (Access: Byte, Half-word, Word)**
- **WTBR2: Address 04A7_H (Access: Byte, Half-word, Word)**

WTBR0								bit
7	6	5	4	3	2	1	0	
–	–	–	D20	D19	D18	D17	D16	
–	–	–	X	X	X	X	X	Initial value
–	–	–	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
RX/WX	RX/WX	RX/WX	R/W	R/W	R/W	R/W	R/W	Attribute

WTBR1								bit
7	6	5	4	3	2	1	0	
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial value
Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

WTBR2								bit
7	6	5	4	3	2	1	0	
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”).

The sub-second registers, WTBR, holds values to be reloaded to the 21-bit down counter. When the 21-bit down counter value becomes “0”, the settings of WTBR are reloaded to the 21-bit down counter.

(See “[49.8. Caution \(Page No.1143\)](#)”).

Remark: The reload value to be set in the sub-second registers corresponds to the time for half a second. One second is reached after counting twice the reload value set in WTBR.

(See “[49.7.1 How to set the count period of 1 second \(Page No.1141\)](#)”).

MB91460 Series**49.4.3 WTHR/WTMR/WTSR: Hour/Minute/Second Registers**

These registers hold time information (HH/MM/SS) for Real-time Clock.

- **WTHR (Hour register): Address 04A8_H (Access: Byte, Half-word)**
- **WTMR (Minute register): Address 04A9_H (Access: Byte, Half-word)**
- **WTSR (Second register): Address 04AA_H (Access: Byte, Half-word)**

WTHR								bit
7	6	5	4	3	2	1	0	
–	–	–	H4	H3	H2	H1	H0	
–	–	–	X	X	X	X	X	Initial value
–	–	–	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
RX/WX	RX/WX	RX/WX	R,W	R,W	R,W	R,W	R,W	Attribute

WTMR								bit
7	6	5	4	3	2	1	0	
–	–	M5	M4	M3	M2	M1	M0	
–	–	X	X	X	X	X	X	Initial value
–	–	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
RX/WX	RX/WX	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

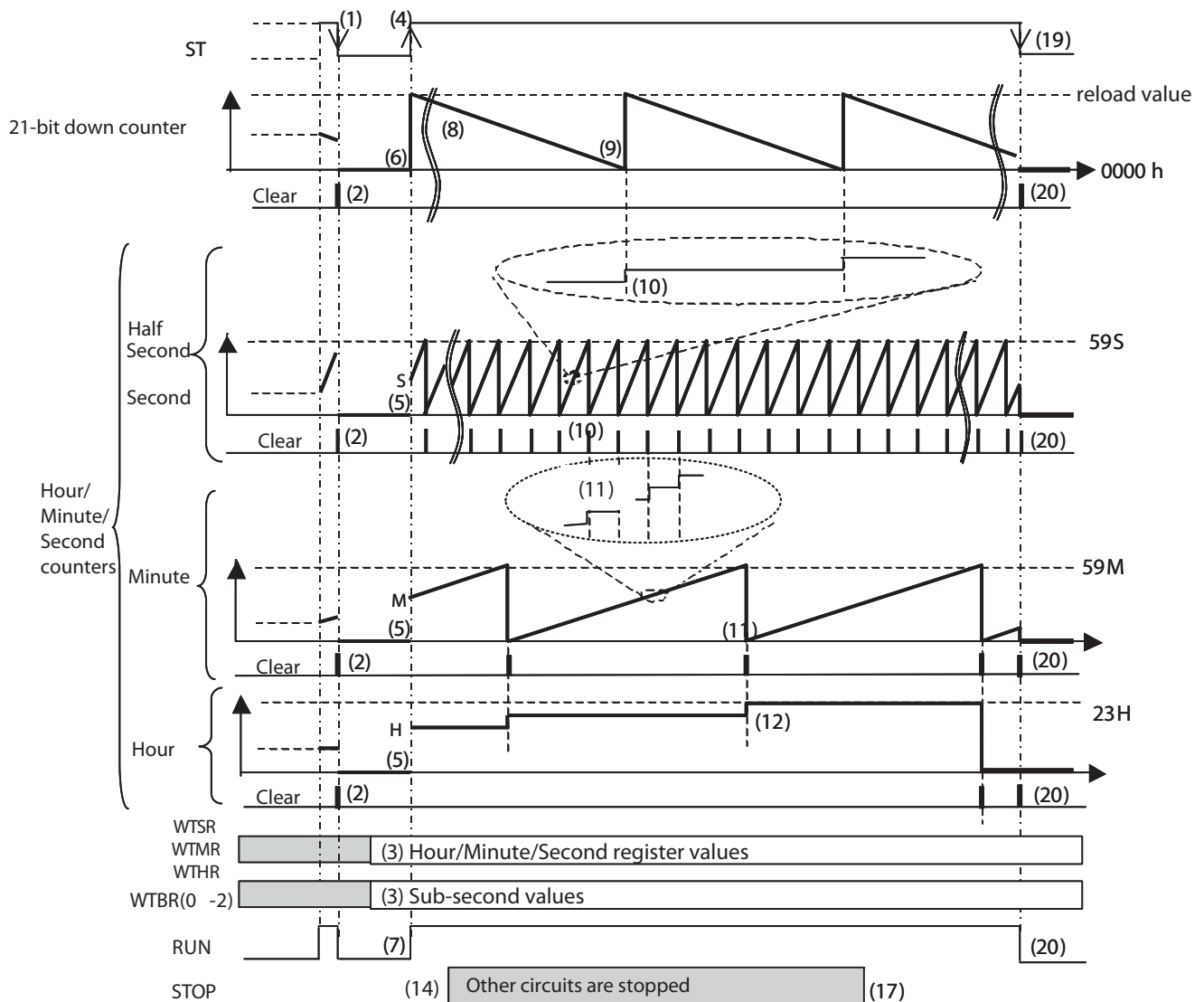
WTSR								bit
7	6	5	4	3	2	1	0	
–	–	S5	S4	S3	S2	S1	S0	
–	–	X	X	X	X	X	X	Initial value
–	–	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	Unchanged	When reset
RX/WX	RX/WX	R,W	R,W	R,W	R,W	R,W	R,W	Attribute

- The values written to the hour/minute/second registers are the initial values to be loaded to the hour/minute/second counters.
By setting “1” to the update bit (WTCR.UPDT) or the start bit (WTCR.ST), the hour/minute/second register values are written to the hour/minute/second counters.
- The hour/minute/second counter values are saved to the hour/minute/second registers every time when the second counter overflows, that is, at intervals of one minute. When the hour/minute/second counters are read, the saved count values, not written ones, are read.
- The hour/minute/second registers consist of two separate sets of registers: one for reading and the other for writing.

(See “49.8. Caution (Page No.1143)”.)

49.5. Operation

Figure 49.5-1 Real-time Clock operation



- (1) The start bit (ST) is set to "1" and then "0". (Register initialization operation)
- (2) This (ST="0") resets to 0 and stops the 21-bit down counter and the hour/minute/second timers.
- (3)
 - The software writes hour, minute, and second values to the hour/minute/second registers, WTHR/WTMR/WTSR.
 - The software writes the appropriate values to the sub-second registers, WTBR0/WTBR1/WTBR2.
 - The interrupt request bits (INT0, INT1, INT2, INT3 and INT4) are initialized, and the interrupt request enable bits (INTE0, INTE1, INTE2, INT3 and INTE4) are set to "interrupts enabled".
- (4) The start bit (ST) is set to "1".
- (5) This (ST="1") causes the values of the hour/minute/second registers, WTHR/WTMR/WTSR, to be loaded to the hour/minute/second timers.
- (6) The values of the sub-second registers, WTBR0/WTBR1/WTBR2, are loaded to the 21-bit down counter.

- (7) The run flag (RUN) is set to “1”.
- (8) The 21-bit down counter begins counting at the CLKMAIN divided by 2 (4/2MHz), CLKSUB divided by 2 (32.768/2kHz) or CLKRC divided by 2 (100/2kHz).
- (9) When the 21-bit down counter reaches “000000_H”, the value of the sub-second registers is loaded to the 21-bit down counter, generating a half-second interrupt request. Each second half-second interrupt a second interrupt will be generated.
- (10) When the second counter counts up to “59”, the counter is cleared next time when the counter counts up, at which the minute counter counts up, generating a 1-minute interrupt request.
- (11) When the minute counter counts up to “59”, the counter is cleared next time when the counter counts up, at which the hour counter counts up, generating a 1-hour interrupt request.
- (12) When the hour counter counts up to “23”, the counter is cleared next time when the counter counts up, at which a 1-day interrupt request is generated.
- (14) The software changes the status of Real-time Clock to STOP. (Set the stop bit (STCR.STOP) to “1”.) Real-time Clock continues to operate in the STOP state.
- (17) The device recovers from STOP state (by interrupt request).
- (20) This (ST=“0”) resets and stops the 21-bit down counter and the hour/minute/second counters.

49.6. Setting

Table 49.6-1 Required Settings to Run Real-time Clock

Setting	Setting Registers	Setting Procedure *
Set a reload value to the sub-second registers.	Sub-second registers (WTBR0, WTBR1 and WTBR2)	See 49.7.1
Initialize Real-time Clock.	RTC control register (WTCR)	See 49.7.2
Set time (hour/minute/second).	Hour/minute/second registers (WTHR/WTMR/WTSR)	See 49.7.3
Activate Real-time Clock.	RTC control register (WTCR)	See 49.7.4

*: For the setting procedure, refer to the section indicated by the number.

Table 49.6-2 Required Settings to Know Time

Setting	Setting Registers	Setting Procedure *
Read time.	Hour/minute/second registers (WTHR/WTMR/WTSR)	See 49.7.6

*: For the setting procedure, refer to the section indicated by the number.

Table 49.6-3 Required Settings to Stop Real-Time Clock

Setting	Setting Registers	Setting Procedure *
Stop Real-time Clock.	RTC control register (WTCR)	See 49.7.7

*: For the setting procedure, refer to the section indicated by the number.

Table 49.6-4 Required Settings to Generate Interrupts to Real-time Clock

Setting	Setting Registers	Setting Procedure *
Set a RTC interrupt vector and RTC interrupt level.	See “Chapter 24 Interrupt Control (Page No.429)” .	See 49.7.10
Enable RTC interrupts. Clear interrupt requests. Enable interrupt requests.	RTC control register (WTCR)	See 49.7.10

*: For the setting procedure, refer to the section indicated by the number.

Table 49.6-5 Required Settings to select clock source to Real-time Clock

Setting	Setting Registers	Setting Procedure *
Set a RTC clock source.	See “13.4.5 CSCFG: Clock Source Configuration Register (Page No.299)” .	See 49.7.11

*: For the setting procedure, refer to the section indicated by the number.

49.7. Q&A

49.7.1 How to set the count period of 1 second

Stop Real-time Clock and then set the sub-second register WTBR. The reload value corresponds to the time needed for half a second, i.e.

- At 32kHz RTC operation set WTBR to “001FFF_H”.
- At 100kHz RTC operation set WTBR to “0061A7_H”.
- At 4MHz RTC operation set WTBR to “0F423F_H”.

49.7.2 How to initialize Real-time Clock

Use the start bit (WTCR.ST).

Changing the start bit from “1” to “0” resets the hour/minute/second counters and the 21-bit down counter to “0” (initialization), and stops count operation.

49.7.3 How to set or update time (hour/minute/second)

Write values to the hour/minute/second registers, WTHR/WTMR/WTSR, and set the update bit (UPDT).

Operation	Update bit (UPDT)
To update the hour/minute/second counters	Set the bit to “1”.

49.7.4 How to start or stop Real-time Clock's counting

Use the start bit (WTCR.ST).

Operation	Start bit (ST)
To stop Real-time Clock's counting	Set the bit to “0”.
To start Real-time Clock's counting	Set the bit to “1”.

49.7.5 How to confirm that Real-time Clock is active

Use the run flag (WTCR.RUN).

Operation	Run flag (RUN)
Real-time Clock is inactive.	The flag is set to “0”.
Real-time Clock is active.	The flag is set to “1”.

49.7.6 How to know the time

Read the hour/minute/second registers, WTHR/WTMR/WTSR.

Note that only byte-access and half-word-access is allowed to these register. So, when these registers are read at a time very close to the changing over the hour or minute boundary as shown below, there is a possibility of misjudging the time. So, read several times to get a logically consistent value.

Example: Read begins at the second register: 02:59:59 (SS) => 03:59:59 (SS) => 03:00:00

Read begins at the hour register: 02:59:59 => 02:00:00 => 03:00:00

49.7.7 How to stop Real-time Clock

See [49.7.4](#)

49.7.8 What are interrupt-related registers?

RTC interrupt vector and level settings.

The following table shows the relationship between interrupt levels and vectors.

For details on interrupt levels and vectors, refer to “[Chapter 24 Interrupt Control \(Page No.429\)](#)”.

Interrupt vectors (Default)	Interrupt level set bit (ICR[4.0])
#132 (0FFDECh)	Interrupt level register ICR58 (047Ah)

The interrupt request flags (INT0,INT1,INT2,INT3 and INT4) are not automatically cleared, so the software must clear them by writing “0” to these flags before control is returned from interrupt processing.

49.7.9 What interrupts are available and how are they selected?

There are four interrupt causes:

Interrupt cause	Interrupt request bit	Interrupt request enable bits
On counting seconds	INT0	INTE0
On counting minutes	INT1	INTE1
On counting hours	INT2	INTE2
On counting days	INT3	INTE3
On counting half-seconds	INT4	INTE4

An interrupt request is made by combination (logical OR) these four interrupt causes. Each cause can be selected with the corresponding interrupt request enable bit.

49.7.10 How to enable interrupts

Use the interrupt request enable bits (WTCR.INTE0, WTCR.INTE1, WTCR.INTE2, WTCR.INTE3 and WTCER.INTE4).

	Setting Procedure
	Interrupt request enable bits (INTE0, INTE1, INTE2, INTE3 and INTE4)
To disable interrupts	Set the bit to “0”.
To enable interrupts	Set the bit to “1”.

To clear interrupt requests,

Use the interrupt request bits (WTCR.INT0, WTCR.INT1, WTCR.INT2, WTCR.INT3 and WTCER.INT4).

	Setting Procedure
	Interrupt request bits (INT0, INT1, INT2, INT3 and INT4)
To clear interrupt requests	Write “0”.

49.7.11 How to select the clock source for RTC

Use the CSC1.CSCFG and CSC0.CSCFG to select the clock source for RTC.

See section [13.4.5 CSCFG: Clock Source Configuration Register](#)

- CSCFG [1:0]: Clock Source Selection for RTC

CSC1-CSC0	Function
00	Real Time Clock is sourced by CLKMAIN
01	Real Time Clock is sourced by CLKSUB
10	Real Time Clock is sourced by CLKRC100kHz
11	Setting prohibited

49.8. Caution

- Setting the interrupt flags (WTCR.INT0, WTCR.INT1, WTCR.INT2, WTCR.INT4 and WTCER.INT4) to “1” due to overflow, and writing “0” to that bit have occurred at the same time, the flag is set to “1”. (Flag setting takes precedence.)
- Writing “1” to the update bit (ETCR.UPDT) and update completion have occurred at the same time, the update bit (UPDT) is set to “0”.
- If the peripheral clock (CLKP) is stopped after updating the hour/minute/second counters using the update bit (WTCR.UPDT), read the hour/minute/second registers to confirm that they have been updated before stopping the peripheral clock.
- When the use of the Real-time Clock module is started, change the start bit (ST) from “1” to “0”, and clear the hour/minute/second counters and the 21-bit down counter to “0”.

- The Sub-second register of the RTC module stores the reload value for the 21-bit prescaler. This value is reloaded after the reload counter reaches "0". When modifying all three bytes, make sure the reload operation will not be performed in between the write instructions. Otherwise the 21-bit prescaler loads the incorrect value of the combination of new data and old data bytes. It is generally recommended that the Sub-Second register is updated while the ST bit is "0".

However, if this update is done immediately after an RTC second interrupt there should be enough time to securely modify the registers until the next reload operation (next second interrupt) even if ST is not set to "0" and the module is in operation.

When updating the registers by using the ST bit the following must be taken into account:

The new value is written into the registers with the rising edge of the RUN bit. This RUN bit is clocked by the RTC clock (32kHz, 100kHz or 4MHz depending on device and mode). To make sure that the update is done properly, write the new values into the registers, set ST to 0, wait for the RUN bit to go low and then start the circuit again by setting ST to 1. RUN will go low at the second rising edge of the RTC clock after ST has been set to 1. It will rise again at the half second rising edge of RTC clock after ST has been set to 1. If this operation is to be done several times directly after each other, wait for RUN to go to high before setting ST to low again.

- If a reload has occurred during updating the sub-second registers, WTBR0 to WTBR2, an unexpected value may be reloaded to the 21-bit down counter. Therefore, it is recommended that the sub-second register, WTBR, should be updated with the start bit (WTCR.ST) set to “0”.
- If all the sub-second registers, WTBR0 to WTBR2, are set to “0”, the 21-bit down counter does not operate, resulting in the Real-time Clock module to be inoperational.
- If a carry has occurred during reading from the hour/minute/second registers, WTHR/WTMR/WTSR, inappropriate values may be read. To avoid this, it is recommended that interrupts (INT0-4) should be used to read time (HH/MM/SS).
- In order for the Real-time Clock module to function properly, the frequency of the Sub clock must be much lower than that of the peripheral clock (CLKP). If not, correct values cannot be read from WTHR/WTMR/WTSR.
- Note that only byte-access is allowed to these register. So, when these registers are read at the very timing of changing over the hour or minute boundary as shown below, there is a possibility of misjudging the time. So, read several times to get a logically consistent value.

Example: Read begins at the second register: 02:59:59=> 03:59:59 => 03:00:00

Read begins at the hour register: 02:59:59 => 02:00:00 => 03:00:00

In this case, the current time should be interpreted as 3 o'clock.

Chapter 50 Clock Calibration Unit

50.1. Overview

The Clock Calibration Module provides possibilities to calibrate the 32kHz CLKSUB or 100kHz CLKRC with respect to the 4MHz clock. This chapter gives an overview of the calibration unit, describes the registers and provides some application notes.

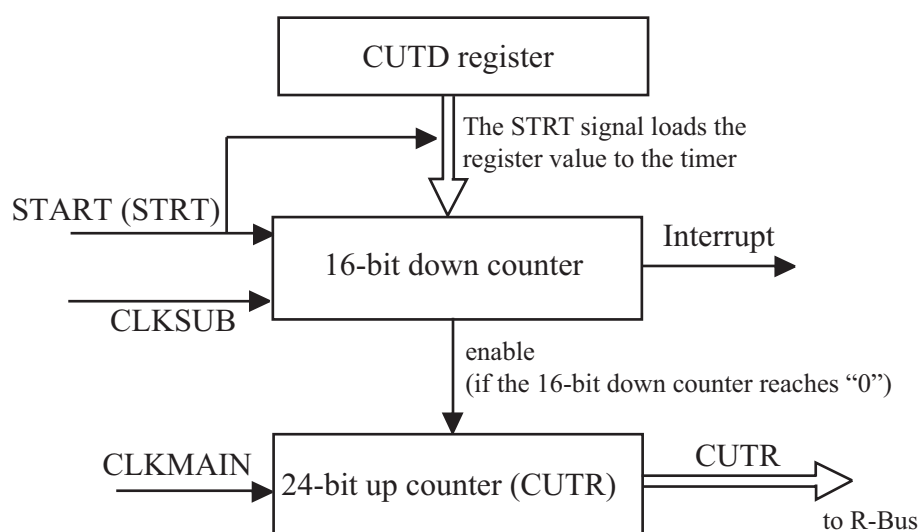
50.1.1 Description

This hardware allows the software to measure time generated by the 32kHz clock (or 100kHz CLKRC) with the 4MHz clock.

By utilizing this hardware in conjunction with software processing, the accuracy of the 32kHz clock (or 100kHz CLKRC) can come closer to that of the 4MHz clock. The measurement result from the Clock Calibration Module can be processed by the software and the setting required for the Real Time Clock Module can be obtained.

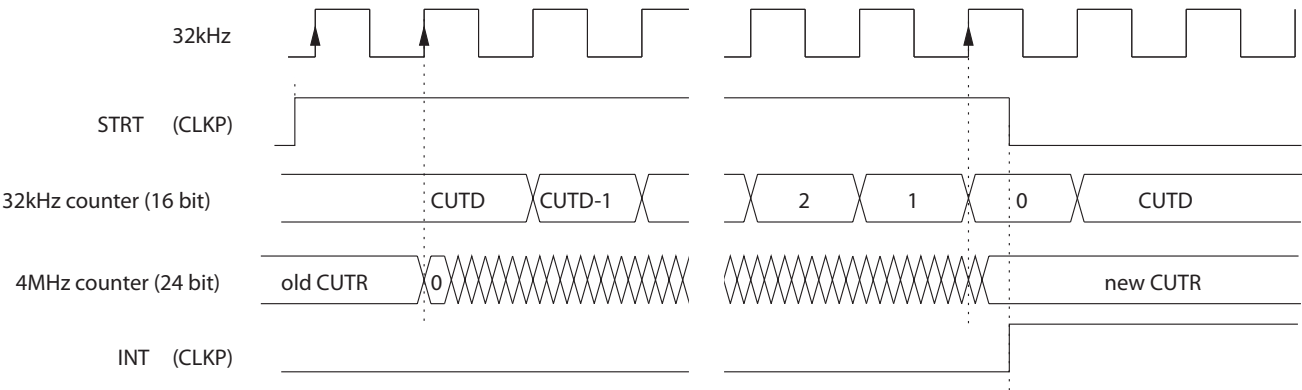
This module consists of two timers, one operating with the 32kHz clock (or 100kHz CLKRC) and the other operating with the 4MHz clock. The 32kHz (100kHz) timer triggers the 4MHz timer and resulting 4MHz timer value is stored in a register. The value stored in this register can be used for the subsequent software processing to calculate the desired Real Time Clock module's setting.

Figure 50.1-1 Block diagram of Sub clock calibration unit



50.2. Timing

Figure 50.2-1 Timing of the measurement process



MB91460 Series

50.3. Clocks

The module operates with 3 different clocks: The 4MHz clock CLKMAIN, the 32kHz clock CLKSUB (or the 100kHz clock CLKRC 100kHz) and the peripheral clock CLKP. Synchronization circuits adapt the different domains.

The clock frequencies have to fulfill the following requirements:

- Clock ratio

$$\frac{1}{CLKSUB} \left(or \frac{1}{CLKRC100kHz} \right) > 2 \times \frac{1}{CLKMAIN} + \frac{3}{CLKP}$$

$$\frac{1}{CLKMAIN} < \frac{1}{2 \times CLKSUB} \left(or \frac{1}{2 \times CLKRC100kHz} \right) - \frac{3}{2} \times \frac{1}{CLKP}$$

$$\frac{1}{CLKP} < \frac{1}{3} \times \frac{1}{CLKSUB} \left(or \frac{1}{CLKRC100kHz} \right) - \frac{2}{3} \times \frac{1}{CLKMAIN}$$

- The input frequencies must not exceed the values given in Table 50.3-1 .

Table 50.3-1 Maximum operation frequencies

	CLKSUB/CLKRC 100kHz		CLKMAIN		CLKP	
maximum	2MHz	500ns	10MHz	100ns	50MHz	20ns

Table 50.3-2 Example of valid clock ratios which fulfill requirements 1 and 2

	CLKSUB		CLKRC 100kHz		CLKMAIN		CLKP	
maximum operation speed	2MHz	500ns	2MHz	500ns	10MHz	100ns	50MHz	20ns
normal operation	32kHz	31.25us	100kHz	10us	4MHz	250ns	>2MHz	500ns

50.4. Register Description

This section lists the registers of the calibration unit and describes the function of each register in detail.

■ Calibration Unit Control Register (CUCR)

Figure 50.4-1 Calibration Unit Control Register (CUCR)

Control Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0004B0H	-	-	-	STRT	-	-	INT	INTEN	CUCRL
Read/write ⇒	(R)	(R)	(R)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ 32kHz/100kHz Timer Data Register (CUTD)

Figure 50.4-2 32kHz/100kHz Timer Data Register (CUTD)

32/100kHz Timer Register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0004B2H	TDD15	TDD14	TDD13	TDD12	TDD11	TDD10	TDD9	TDD8	CUTDH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

32/100kHz Timer Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0004B3H	TDD7	TDD6	TDD5	TDD4	TDD3	TDD2	TDD1	TDD0	CUTDL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

See also [13.4.5 CSCFG: Clock Source Configuration Register \(Page No.299\)](#) for the configuration of the Calibration Unit clock sources.

■ 4MHz Timer Data Register (CUTR1/CUTR2)

Figure 50.4-3 4MHz Timer Data Register (CUTR1/CUTR2)

4MHz Timer Register1 high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0004B4H	-	-	-	-	-	-	-	-	CUTR1H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register1 low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0004B5H	TDR23	TDR22	TDR21	TDR20	TDR19	TDR18	TDR17	TDR16	CUTR1L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register2 high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0004B6H	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	TDR9	TDR8	CUTR2H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register2 low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0004B7H	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	CUTR2L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Table 50.4-1 Calibration Unit registers

Address	Register				Block
	+0	+1	+2	+3	
0004B0 _H	CUCR [R/W] ----- --0--00		CUTD [R/W] 10000000 00000000		Calibration Unit of 32kHz and 100kHz oscillator
0004B4 _H	CUTR1 [R] ----- 00000000		CUTR2 [R] 00000000 00000000		

50.4.1 Calibration Unit Control Register (CUCR)

The Control Register (CUCR) has the following functions:

- start / stop calibration measurement
- enable / disable interrupt request
- indicates the end of the calibration measurement

Figure 50.4-4 Calibration Unit Control Register (CUCR)

Control Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0004B1H	-	-	-	STRT	-	-	INT	INTEN	CUCRL
Read/write ⇒	(R)	(R)	(R)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BIT[0]: INTEN - Interrupt request enable

0	interrupt request disabled (default)
1	interrupt request enabled

This is the interrupt request enable bit corresponding to the INT bit. When this bit is set to 1 and the INT flag is set by the hardware, the calibration module signals an interrupt request to the CPU. The INT-flag itself is not affected by the INTEN bit and is set by hardware even if interrupt requests are disabled (INTEN=0).

BIT[1]: INT - Interrupt

0	calibration ongoing / module inactive (default)
1	calibration completed

This flag indicates the end of the calibration. When the 32kHz/100kHz timer reaches zero after the start of calibration, the 4MHz Timer Data Register stores the last 4MHz timer value and the INT flag is set to 1.

The read-modify-write operation to this flag results in reading 1 and writing 0 to this flag clears this flag (INT=0). Writing 1 to this bit has no effect.

The interrupt flag INT is not reset by hardware. Therefore, it must be reset by software before starting a new calibration. Otherwise the end of the calibration process is only signaled by the STRT bit (the INT flag stays 1 also during calibration).

BIT[4]: STRT - Calibration Start

0	calibration stopped, module switched off (default)
1	start calibration

When the STRT bit is set to 1 by the software, the calibration starts. The 32kHz/100kHz Timer starts counting down from the value stored in the 32kHz/100kHz Timer Data Register and the 4MHz Timer starts counting up from zero.

When the 32kHz/100kHz Timer reaches zero, this bit is reset to 0 by the hardware.

If 0 is written into this bit by the software during the calibration process, the calibration is immediately stopped. If writing 0 by the software and reset to 0 by the hardware happens at the same time, the hardware operation supersedes the software operation. This means the calibration is properly completed and the INT bit is set to "1". Writing 1 to this bit during the calibration has no effect.

50.4.2 32kHz / 100kHz Timer Data Register (16 bit) (CUTD)

The 32kHz/100kHz Timer Data Register (CUTD) holds the value which determines the duration of calibration (32kHz/100kHz reload value).

Figure 50.4-5 32kHz / 100kHz Timer Data Register (16 bit) (CUTD)

32/100kHz Timer Register high byte	15	14	13	12	11	10	9	8	← Bit no.
Address : 0004B2H	TDD15	TDD14	TDD13	TDD12	TDD11	TDD10	TDD9	TDD8	CUTDH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

32/100kHz Timer Register low byte	7	6	5	4	3	2	1	0	← Bit no.
Address : 0004B3H	TDD7	TDD6	TDD5	TDD4	TDD3	TDD2	TDD1	TDD0	CUTDL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Default value is 0x8000 which corresponds to a measurement duration of 1 second, if a 32.768kHz crystal is used.

This register should be written only when the calibration is inactive (STRT=0).

The 32kHz/100kHz Timer Register stores the value to specify the duration of the calibration. When the calibration is started, the stored value is loaded into the 32kHz/100kHz Timer and the timer starts counting down until it reaches zero.

If CUTD is initialized with 0000 an underflow will occur and the measurement will take (FFFF hex + 1) x 1/CLKSUB (or: 1/CLKRC 100kHz)

The 32kHz/100kHz timer operates with the 32kHz or with the 100kHz clock (see chapter “[13.4.5 CSCFG: Clock Source Configuration Register \(Page No.299\)](#)” how to select between those clocks.

In order to achieve a measurement duration of 1 second at a 32kHz clock, the CUTD register has to be load with 0x8000 = 32768 dec. This number results from the exact clock frequency of the crystal, which is CLKSUB=32768 Hz. The ideal values of the measurement results (if a 4.00MHz crystal is used) are shown in the following table.

In order to achieve a measurement duration of half a second at a 100kHz clock, the CUTD register has to be load with 0xC350 = 50000 dec. This number results from the exact clock frequency of the crystal, which is CLKRC 100kHz=100000 Hz. The ideal values of the measurement results (if a 4.00MHz crystal is used) are shown in the following table.

Table 50.4-2 32kHz : Ideal measurement results depending on measurement duration

duration of calibration	CUTD value	CUTR value
2 sec	0x0000	0x7A1200
1.75 sec	0xE000	0x6ACFC0
1.5 sec	0xC000	0x5B8D80
1.25 sec	0xA000	0x4C4B40
1 sec	0x8000	0x3D0900
0.75 sec	0x6000	0x2DC6C0
0.5 sec	0x4000	0x1E8480
0.25 sec	0x2000	0x0F4240

The duration of the whole process from writing a 1 into the STRT bit until STRT is reset by hardware is longer than the actual calibration measurement time, due to synchronization between the different clock domains. $\text{Process Duration} < (\text{CUTD} + 3) \times 1/\text{CLKSUB}$.

The calibration measurement time is exact $\text{CUTD} \times \text{CLKSUB}$.

Table 50.4-3 100kHz : Ideal measurement results depending on measurement duration

duration of calibration	CUTD value	CUTR value
0.5 sec	0xC350	0x1E8480
0.25 sec	0x61A8	0x0F4240
0.125 sec	0x30D4	0x07A120
0.1 sec	0x2710	0x061A80

The duration of the whole process from writing a 1 into the STRT bit until STRT is reset by hardware is longer than the actual calibration measurement time, due to synchronization between the different clock domains. $\text{Process Duration} < (\text{CUTD} + 3) \times \text{CLKRC } 100\text{kHz}$.

The calibration measurement time is exact $\text{CUTD} \times \text{CLKRC } 100\text{kHz}$.

50.4.3 4MHz Timer Data Register (24 bits) (CUTR)

The Timer Data Register (CUTR) holds the value of the calibration result (4MHz counter)

Precaution: Reading this register during calibration, results in random values.

The end of calibration is indicated by the INT-bit and the STRT-bit in the CUCR-register.

After INT has changed from 0 to 1 / STRT has changed from 1 to 0, the value of CUTR is valid.

Figure 50.4-6 4MHz Timer Data Register (24 bits) (CUTR)

4MHz Timer Register1 high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0004B4H	-	-	-	-	-	-	-	-	CUTR1H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register1 low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0004B5H	TDR23	TDR22	TDR21	TDR20	TDR19	TDR18	TDR17	TDR16	CUTR1L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register2 high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0004B6H	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	TDR9	TDR8	CUTR2H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Register2 low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0004B7H	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	CUTR2L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

The 4MHz Timer Data Register stores the result of the calibration. When the calibration is started, the 4MHz Timer starts counting up from zero. When the 32kHz/100kHz Timer reaches zero, the 4MHz Timer stops counting and the register holds the calibration result until the next calibration is triggered by software.

Reading this register during calibration, results in random values.

The end of calibration is indicated by the INT-bit and the STRT-bit in CUCR-register. After these bits changed from 0 to 1, resp. 1 to 0, the value of CUTR is valid.

Writing into this register by software has no effect.

The 4MHz Timer operates with the 4MHz clock.

50.5. Application Note

This section lists application notes concerning accuracy of the calibration, power dissipation and measurement duration.

50.5.1 32kHz setting

The setting of the 32kHz Timer Data Register can be calculated in the following way.

If the duration of 1 second is desired for the calibration, 8000Hex = 32768Dec should be set in the 32kHz Timer Data Register and it represents 32,768 pulses of the 32.768kHz clock.

This setting should result in the stored value of approximately 3D0900Hex in the 4MHz Time Data Register. This value represents 4,000,000 pulses of the 4MHz oscillator.

50.5.2 100kHz setting

The setting of the 100kHz Timer Data Register can be calculated in the following way.

If the duration of 0.5 second is desired for the calibration, C350Hex = 50000Dec should be set in the 100kHz Timer Data Register and it represents 50,000 pulses of the 100kHz clock.

This setting should result in the stored value of approximately 1E8480Hex in the 4MHz Time Data Register. This value represents 2,000,000 pulses of the 4MHz oscillator.

50.5.3 Measurement Results

Table 50.5-1 Ideal measurement results (CUTR) with 32.768kHz and 4.0MHz oscillators

duration of calibration	CUTD value	CUTR value
2 sec	0x0000	0x7A1200
1.75 sec	0xE000	0x6ACFC0
1.5 sec	0xC000	0x5B8D80
1.25 sec	0xA000	0x4C4B40
1 sec	0x8000	0x3D0900
0.75 sec	0x6000	0x2DC6C0
0.5 sec	0x4000	0x1E8480
0.25 sec	0x2000	0x0F4240

The key to the use of the calibration module is power dissipation as well as accuracy of the calibration.

Table 50.5-2 Ideal measurement results (CUTR) with 100kHz and 4.0MHz oscillators

duration of calibration	CUTD value	CUTR value
0.5 sec	0xC350	0x1E8480
0.25 sec	0x61A8	0x0F4240
0.125 sec	0x30D4	0x07A120
0.1 sec	0x2710	0x061A80

The key to the use of the calibration module is power dissipation as well as accuracy of the calibration.

50.5.4 Accuracy:

The accuracy of the calibration is dependent on the clock frequency used by the 4MHz Timer and duration of the calibration. The maximum error of the 4MHz timer is +/- 1 digit. If the clock frequency is 4MHz and duration of the calibration is 1 second, the achieved accuracy is calculated in the following way:

$$0.25\mu\text{s (Clock cycle time)} / 1 \text{ second (duration)} = 0.25 \text{ ppm.}$$

In general:

$$\text{Accuracy} = (\text{Clock cycle time of 4MHz Timer}) / (\text{Duration of calibration})$$

50.5.5 Power dissipation:

Suppose the current consumption I_{RUN} in RUN state is 20 times the consumption I_{RTC} in STOP state with RTC running ($I_{\text{RUN}} = 20 \times I_{\text{RTC}}$).

If the MCU is woken up from STOP state with RTC running by software to trigger the calibration measurement every minute and the duration of the calibration is set to 1 second, the increase in the power dissipation can be $20 \times I_{\text{RTC}} / 60 = 1/3 \times I_{\text{RTC}}$.

Therefore the software has to make sure that the increase should not affect the hardware limitations coming from the system requirements. For example, the software has to be designed to trigger the calibration least frequently.

It is generally recommended that the theoretical increase in power dissipation is not more than 5% particularly in the STOP state with RTC running of the MCU.

50.5.6 Measurement limits:

The limit to the duration of the calibration is roughly 2 seconds if the 32kHz/100kHz Timer is operating with 32kHz clock (0.5 seconds if operating with 100kHz). On the other hand, the 4MHz Timer can measure time up to 4 seconds if it is working with a 4MHz clock.

Chapter 51 Low Voltage Reset/Interrupt (Supply Supervisor)

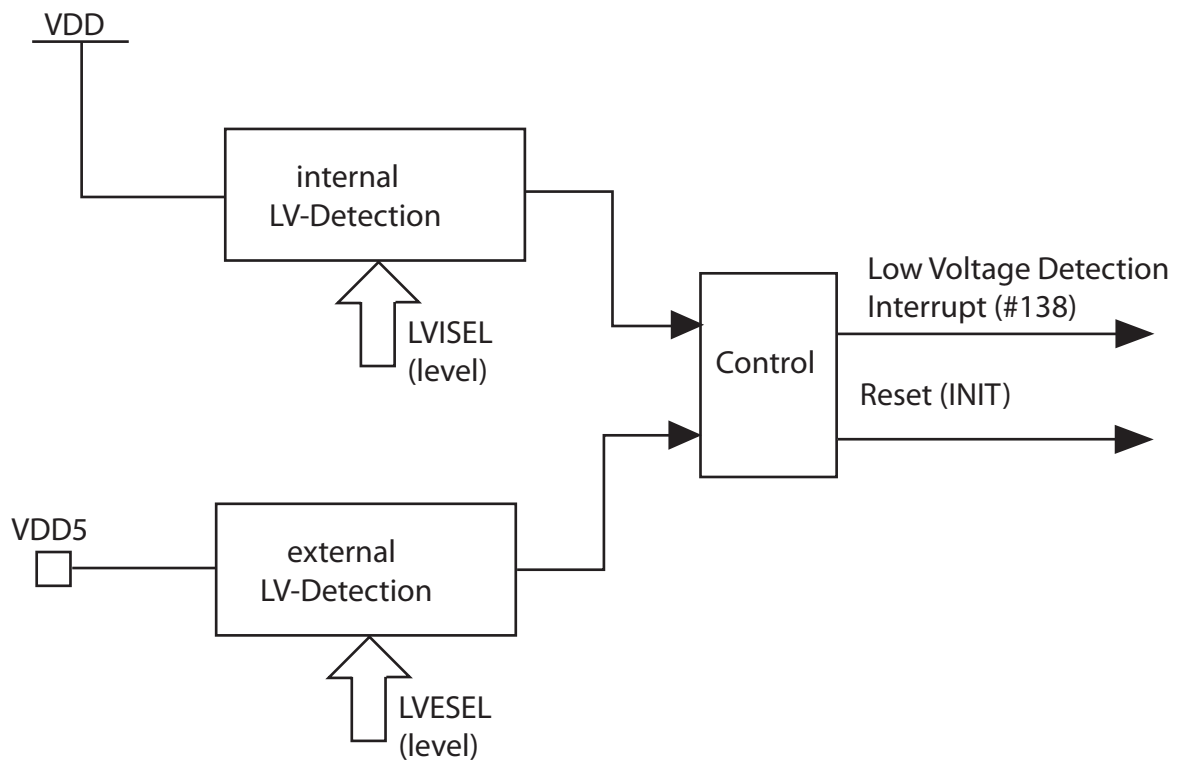
51.1. Overview

- Supply supervisor for generating a low voltage reset or interrupt depending on the supply state of either the internal or external supply voltage.

51.2. Features

- Generates a low voltage reset or a low voltage interrupt
- Interrupt activation source can be selected between external supply (VDD5) detection and internal regulator output VDD detection. [See "Block diagram of regulator control" on P. 1161](#)
- Selectable trigger levels for external and internal supply levels
- Power down function

Figure 51.2-1 Block diagram of the Low Voltage Reset/Interrupt



51.3. Registers

51.3.1 LV Detection Control Registers

Controls the low voltage detection function and indicates it's status.

- **LVDET: Address 04C5h (Access: Byte, Halfword, Word)**

7	6	5	4	3	2	1	0	bit
-	LVSEL	LVEPD	LVIPD	LVREN	-	LVIE	LVIRQ	
-	0	0	0	0	-	0	0	Initial value (INITX pin input, watchdog reset)
-	X	X	X	X	-	0	0	Initial value (Software reset)
R0/W0	R/W	R/W	R/W	R/W	R0/W0	R/W	R/W/RM1	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7: Reserved bit. The read value is always '0'.
- Bit6: Low-Voltage Interrupt Source selection.

LVSEL	Function
0	Internal (VDD) LV detection used for LV-Int and LV-Reset [Initial value]
1	External (VDD5) LV detection used for LV-Int and LV-Reset

- Bit5: External Low-Voltage Detection Power-Down.

LVEPD	Function
0	External (VDD5) LV detection active [Initial value]
1	External (VDD5) LV detection power down

- Bit4: Internal Low-Voltage Detection Power-Down.

LVIPD	Function
0	Internal (VDD) LV detection active [Initial value]
1	Internal (VDD) LV detection power down

- Bit3: Low-Voltage Reset Enable.

LVREN	Function
0	Low Voltage Reset Disabled [Initial value]
1	Low Voltage Reset Enabled

- Bit2: Reserved bit.
- Bit1: Low-Voltage Interrupt Enable.

LVIE	Function
0	Low Voltage Interrupt Disabled [Initial value]
1	Low Voltage Interrupt Enabled

- Bit0: Low-Voltage Interrupt Flag.

LVIRQ	Function
0	Low Voltage Interrupt Flag (no interrupt request) [Initial value]

1	Low Voltage Interrupt Flag (interrupt requested)
---	--

• **LVSEL: Address 04C4h (Access: Byte, Halfword, Word)**

7	6	5	4	3	2	1	0	bit
LVSEL3	LVSEL2	LVSEL1	LVSEL0	LVISEL3	LVISEL2	LVISEL1	LVISEL0	
0	0	0	0	0	1	1 / 0 *1	1	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

1. Initial values of LVISEL1 is 0 on some devices, see the table below.

(See "Meaning of Bit Attribute Symbols (Page No.15)" for details of the attributes.)

• Bit7-4: External LV detection voltage level

LVSEL3-LVSEL0	Trigger level
1001	4.2V +/- 0.1V
1000	4.1V +/- 0.1V
0111	4.0V +/- 0.1V
0110	3.9V +/- 0.1V
0101	3.8V +/- 0.1V
0100	3.7V +/- 0.1V
0011	3.6V +/- 0.1V
0010	3.2V +/- 0.1V
0001	3.0V +/- 0.1V
0000	2.8V +/- 0.1V (initial)

• Bit3-0: Internal LV detection voltage level

LVISEL3-LVISEL0	Trigger level	Comments
0111	1.6V +/- 0.1V	
0110	1.5V +/- 0.1V	Initial value on all devices which are not listed for 1.6V below.
0101	1.4V +/- 0.1V	
0100	1.3V +/- 0.1V	Initial value on MB91F464Ax, MB91F465Kx, MB91F467BB, MB91F467Dx, MB91F467RC, MB91F467M, MB91F469Gx)
0011	1.2V +/- 0.1V	
0010	1.1V +/- 0.1V	
0001	1.0V +/- 0.1V	
0000	0.9V +/- 0.1V	

Note: The set level of the Internal Low Voltage Detection is only effective if the Main Regulator is switched off (since the sub-regulator output level can be configured from 1.2 - 1.9 V, the trigger level must also be configurable). Otherwise (with main regulator on) the default level is applied internally by hardware to the low voltage detection module (the register setting is not changed in this case and will be applied after the main regulator is switched off). See "Regulator Control" on P. 1161

Chapter 52 Regulator Control

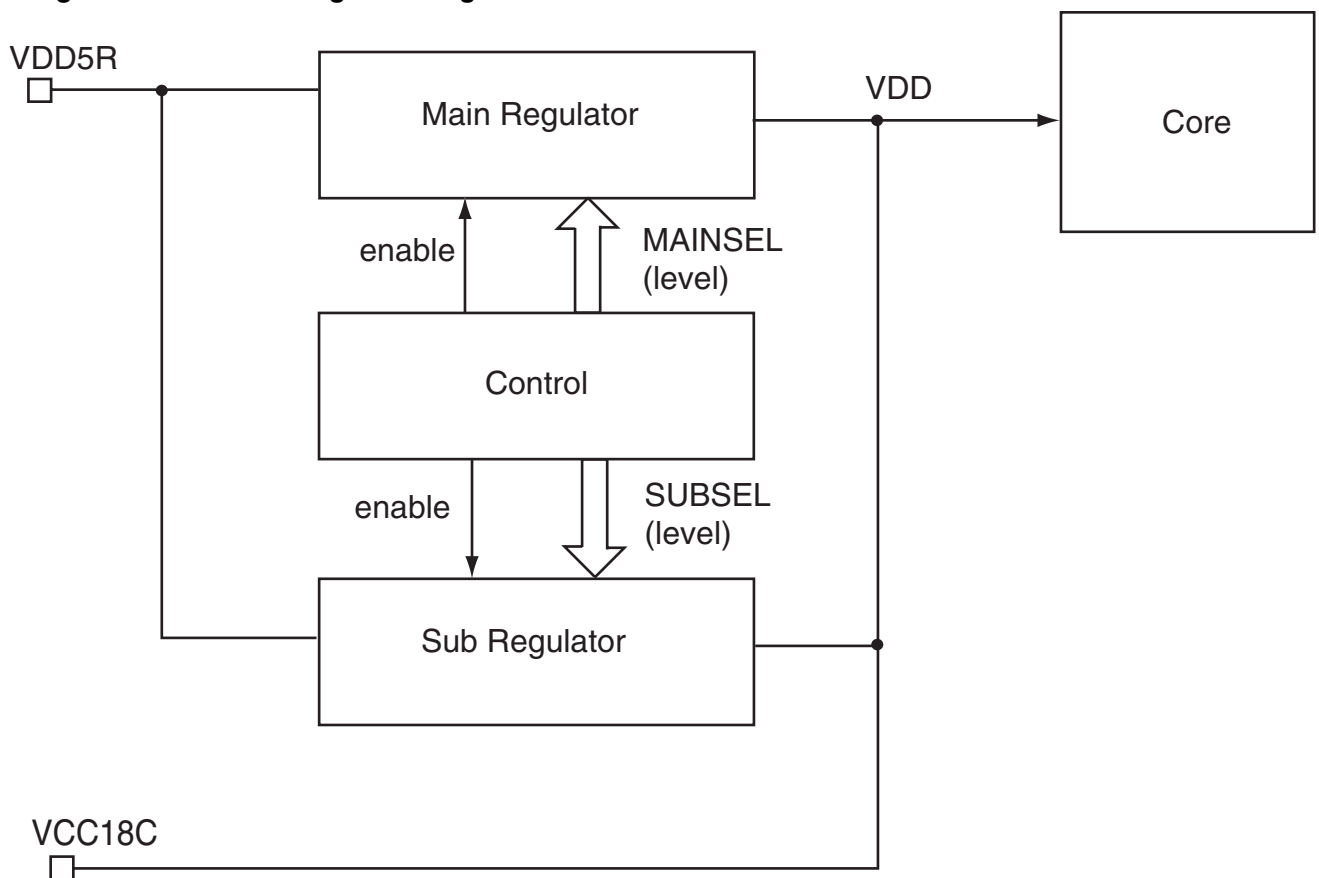
52.1. Overview

- Module for controlling the behaviour of the MAIN-Regulator and SUB-Regulator in the device modes.

52.2. Features

- The Main Regulator can be enabled or disabled during Sub-RUN and STOP/STOP with RTC running.
- Main regulator standby flag output
- Selectable Sub regulator output voltage for Sub-RUN and STOP/STOP with RTC running

Figure 52.2-1 Block diagram of regulator control



52.3. Registers

52.3.1 Regulator Control Registers

Controls the regulator function.

- **REGCTR: Address 04CFh (Access: Byte, Halfword, Word)**

7	6	5	4	3	2	1	0	bit
-	-	-	MSTBO	-	-	MAINKPEN	MAINDSBL	
X	X	X	X	X	X	0	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R0/WX	R0/WX	R0/WX	R	R0/WX	R0/WX	R/W	R/W	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-5: Reserved bit. The read value is always “0”.
- Bit4: Main regulator Standby output flag.

MSTBO	Function
0	Main regulator is in RUN mode
1	Main regulator is in STANDBY mode

- Bit3-2: Reserved bit. The read value is always “0”.
- Bit1: Main Regulator enable in STOP state (with and without running RTC).

MAINKPEN	Function
0	Main regulator disabled in STOP state (with and without running RTC) [Initial value]
1	Main regulator enabled in STOP state (with and without running RTC)

- Bit0: Main Regulator disable in Sub-RUN state.

MAINDSBL	Function
0	Main regulator enabled in Sub-RUN state [Initial value]
1	Main regulator disabled in Sub-RUN state

MB91460 Series

- **REGSEL: Address 04CEh (Access: Byte, Halfword, Word)**

7	6	5	4	3	2	1	0	bit
-	-	FLASHSEL	MAINSEL	SUBSEL 3	SUBSEL 2	SUBSEL 1	SUBSEL 0	
0	0	0	0	0	1	1 / 0 *1	0	Initial value (INITX pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R0/WX	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

1. Initial values of SUBSEL1 is 0 on some devices, see the table below.

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

- Bit7-6: Reserved bit. The read value is always “0”.
- Bit5-4: Flash memory supply mode and Main Regulator supply mode

FLASHSEL	MAINSEL	Function
0	0	Flash memory and Main Regulator operation mode is 1.8V [Initial value]
1	1	Flash memory and Main Regulator operation mode is 1.9V
other		prohibited

- Bit3-0: Sub-regulator voltage level

SUBSEL3-SUBSEL0	Voltage level	Comments
0111	1.9V +/- 0.1V	
0110	1.8V +/- 0.1V	Initial value on all devices which are not listed for 1.6V below.
0101	1.7V +/- 0.1V	
0100	1.6V +/- 0.1V	Initial value on MB91F464Ax, MB91F465Kx, MB91F467BB, MB91F467Dx, MB91F467RC, MB91F467M, MB91F469Gx)
0011	1.5V +/- 0.1V	
0010	1.4V +/- 0.1V	
0001	1.3V +/- 0.1V	
0000	1.2V +/- 0.1V	

Note: The set level of the Sub-regulator voltage is only effective if Main Regulator is switched off. Otherwise (with main regulator on) the default level is applied internally by hardware to the Sub-Regulator (the register setting is not changed in this case and will be applied after the main regulator is switched off).

		Regulator Control							
Device Mode	Sub Mode	Used Regulator	Main Regulator			switch by	Sub Regulator		
			1.8V	switch by MAIN_SEL[]	1.9V		1.8V	switch by SUB_SEL[]	1.2V
RESET	INIT	MAIN	on (def)		on		off		off
RUN	PLL	MAIN	on (def)		on		off		off
	2MHz	MAIN	on (def)		on		off		off
SUB RUN	32kHz	MAIN	on (def)	FLASH Active	on	MAINDSBL	on		on
	RC 100kHz	or SUB	on (def)		on	MAINDSBL	on		on
	RC 2MHz	MAIN	on (def)		on		off		off
SLEEP	PLL	MAIN	on (def)		on		off		off
	32kHz	MAIN	on (def)		on		off		off
SUB SLEEP	32kHz	MAIN	on (def)		on	MAINDSBL	on		on
	RC 100kHz	or SUB	on (def)		on	MAINDSBL	on		on
	RC 2MHz	MAIN	on (def)		on		off		off
RTC	4MHz	SUB	on		on	MAINKPEN	on (def)		on
	32kHz	SUB	on		on	MAINKPEN	on (def)		on
	RC 100kHz	SUB	on		on	MAINKPEN	on (def)		on
	RC 2MHz	SUB	on		on	MAINKPEN	on (def)		on
STOP	STOP	SUB	on		on	MAINKPEN	on (def)		on
	STOP+HIZ	SUB	on		on	MAINKPEN	on (def)		on

Chapter 53 Fixed Mode-Reset Vector / Boot-ROM

53.1. Overview

The Boot ROM is a fixed start-up routine, which is located at memory addresses 0xB000 to 0xBFFF. The entry point 0xBFF8 is determined by the Fixed Reset Vector if the device is configured with the mode pins set to MD[2:0]="000" (internal ROM/vector mode). In this mode MB91460 series devices use the Fixed Mode Vector data (FMV, address 0x0F:FFF8, data 0x06000000) and Fixed Reset Vector data (FRV, address 0x0F:FFFC, data 0x0000BFF8). The data of both these vectors are independent from the flash content at these two addresses.

The purpose of the Boot ROM is to configure the device after a reset and to provide a simple serial bootloader for programming the embedded flash memories.

Therefore it is executed always after the Reset Cancellation Sequence (see chapter 9 [Section 9.5.6 Reset Cancellation Sequence \(Page No.235\)](#) or [Section 9.7.6 Reset Cancellation Sequence \(Page No.237\)](#)) of every INIT or RST reset.

53.2. Check for Boot Conditions

The check for boot conditions is slightly different for the evaluation chip MB91V460A (A-version) and the flash derivatives (e.g. MB91F467DA).

53.2.1 Evaluation Chip MB91V460A¹

After the chip initialization and saving the RSRR (Reset Cause Register) to CPU register R4, the Boot Security Vector (BSV: Vector #144, 0x0F:FDBC) will be checked. This check is performed as follows: if the data of this vector represents a valid address in the specified address range, the Boot Security Vector itself becomes valid.

Device	Valid Boot Security Vector address range
MB91V460A	0x04:0000 – 0x13:FFFF

If the Boot Security Vector is valid, the Boot ROM is left and the user application is started at the address given by this vector.

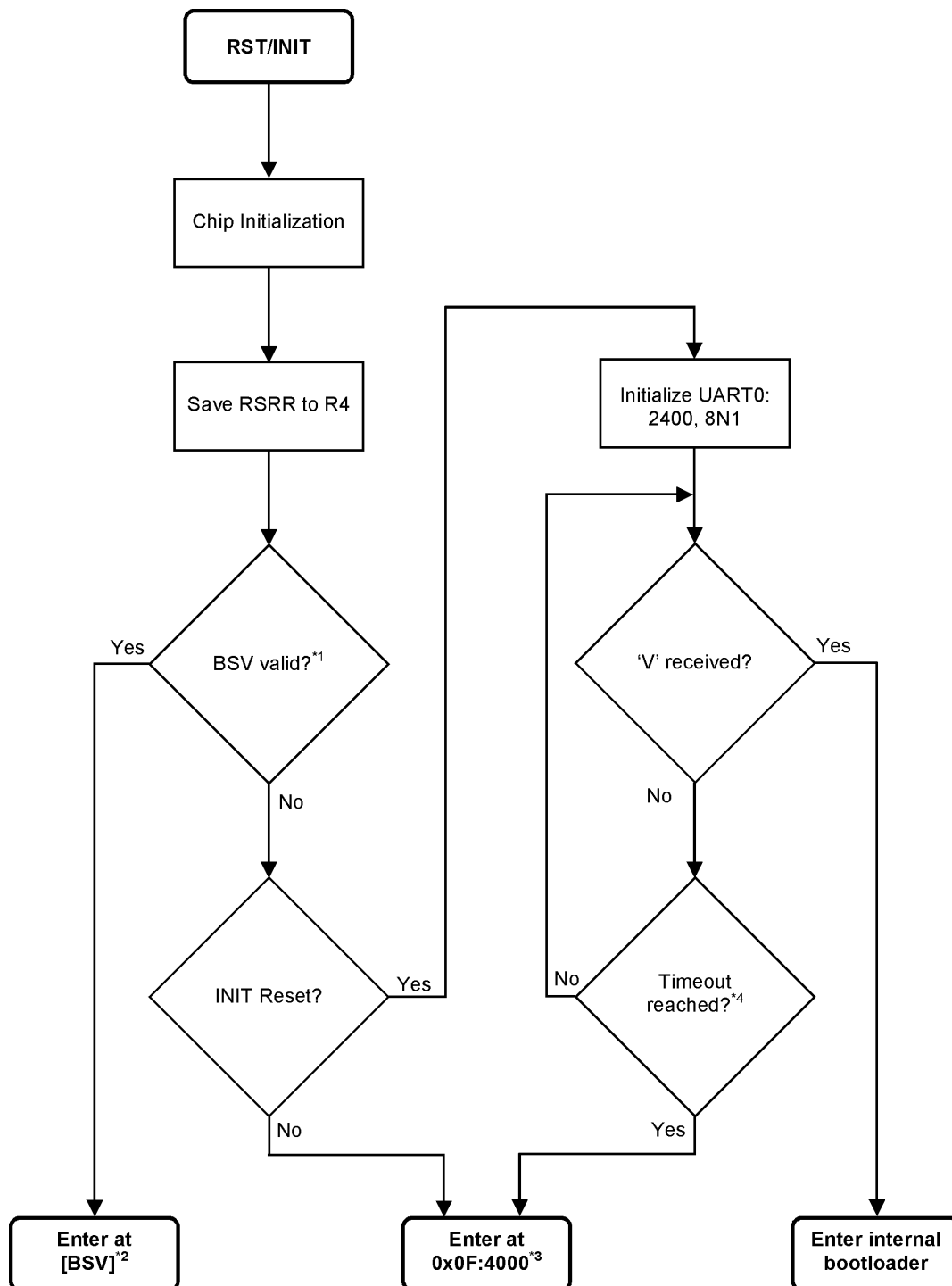
The purpose of this feature is to disable the execution of the internal bootloader due to security reasons or to minimize startup time of the application. Only if the user sets the Boot Security Vector to an address outside the given address range or leaves this vector at default content after erase (0xFFFF:FFFF) the internal bootloader can be entered.

If the Boot Security Vector is not valid, the reset cause will be checked as second boot condition. Only if the reset cause was an INIT reset (external INITX pin input, RSRR=0x80), the check for boot conditions continues. Otherwise Boot ROM is left and application is started at default user program entry address 0x0F:4000.

If the reset cause was an INIT reset, UART0 is initialized: 2400 baud, 8 data bits, 1 stop bit, no parity. UART-reception is checked for about 100 ms. If during this time period the ASCII-character "V" (0x56) is received, the internal bootloader is entered. Otherwise Boot ROM is left and application is started at default user program entry address 0x0F:4000.

1. The new evaluation chip, MB91FV460B, has a separate flash memory instead of the Boot ROM. This flash can be programmed with the Boot ROMs content of the derivate to be emulated.

Figure 53.2-1 Flow Chart of checking boot conditions on MB91V460A



*1) Boot-Security-Vector points to address in valid address range (0x04:000 – 0x13:FFFF)?

*2) Start user application at address given by Boot Security Vector

*3) Start user application at default user program entry address

*4) Timeout about 100 ms

53.2.2 Flash devices of MB91460 series (MB91F46x)

After the chip initialization and saving the RSRR (Reset Cause Register) to CPU register R4, there is a check for boot conditions. All Flash devices have two Boot Security Vectors (BSV1: 0x14:8004, BSV2: 0x14:800C). These vectors are located in parallel sector to the Flash Security Vectors (FSV1, FSV2):

	<i>Sector SA4 (8kB)</i>	<i>Sector SA5 (8kB)</i>

0x14:8008	FSV2	BSV2
0x14:8000	FSV1	BSV1
	64bit width	

The Flash Security Vectors (FSV1, FSV2) are used for configuring the protection mode of the flash memory sectors and do not influence startup of Boot ROM. Refer to [Chapter 55 Flash Security \(Page No.1205\)](#).

At first, BSV1 is checked: if the data of this vector represents a valid address in the specified address range (depending on Flash-ROM size), the Boot Security Vector itself becomes valid.

Table 53.2-1 Boot security vector address ranges

Device	Valid Boot Security Vector address range
MB91F464Ax	0x0A:0000 – 0x0F:FFFF 0x14:8000 – 0x14:FFFF
Devices with 288 or 416 or 544 KByte flash memory	0x08:0000 – 0x0F:FFFF 0x14:8000 – 0x14:FFFF
Devices with 832 or 1088 KByte flash memory	0x04:0000 – 0x14:FFFF
Devices with 2112 KByte flash memory	0x04:0000 – 0x24:FFFF

If BSV1 is valid, there will be an additional check before entering user program at the entry address given by BSV1 **(1)**. Otherwise checks for entering the internal bootloader will be done **(2)**.

The purpose of this feature is to disable the execution of the internal bootloader due to security reasons or to minimize startup time of application. If the user sets BSV1 to a valid address range, this bootloader cannot be entered any more.

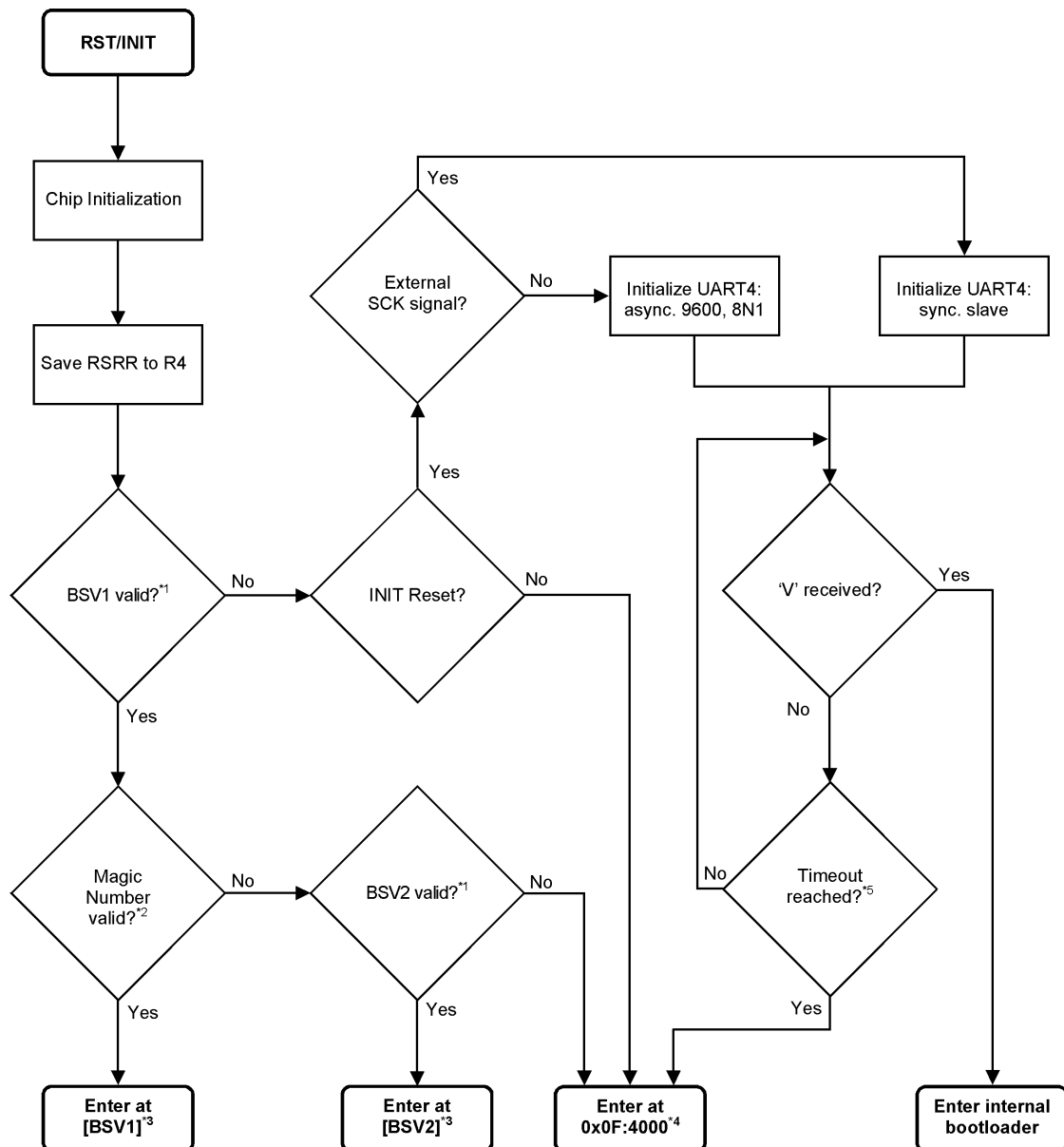
(1) If the check for BSV1 is valid, the Magic Number, which should be located on the four bytes before the address BSV1 points to, is compared to **0x000A897A**. If the Magic Number matches this value, the user application is entered at the address given by BSV1.

The Magic Number is used as flag for a valid user application, or especially for a user bootloader. If this user bootloader should be reprogrammed, a second user bootloader, which handles the re-programming of the first user bootloader, has to be located at the address BSV2 points to. If BSV2 does not point to a valid address range, then application is started at default user program entry address 0x0F:4000. For more information on this bootloader update strategy refer to chapter [53.5. Bootloader Update Strategy \(Page No.1171\)](#).

(2) If the check for BSV1 is not valid, the reset cause will be checked as second boot condition. Only if the reset cause was an INIT reset (external INITX input, RSRR=0x80), the check for boot conditions continues. Otherwise Boot ROM is left and application is started at default user program entry address 0x0F:4000.

If the reset cause was an INIT reset, serial clock pin SCK4 is checked for an external clock signal. Therefore logic level at this pin is monitored for about 1ms. If port level is constant, UART4 is initialized to asynchronous mode: 9600 baud, 8 data bits, 1 stop bit, no parity. If port level changes, UART4 is initialized to synchronous slave mode. UART-reception is checked for about 100 ms. If during this time period the ASCII-character "V" (0x56) is received, the internal bootloader is entered. Otherwise Boot ROM is left and application is also started at default user program entry address 0x0F:4000.

Figure 53.2-2 Flow Chart of checking boot conditions on flash derivatives of MB91460 series



*1) Boot Security Vector points to address in Flash-ROM

*2) Magic Number = 0x0A897A?

*3) Start user application at address given by Boot Security Vector 1/2

*4) Start user application at default user program entry address

*5) Timeout about 100 ms

53.2.3 Internal Bootloader Description

If a valid boot condition for entering the internal bootloader was met, the ASCII-character “F” (0x46) will be transmitted via UART0 (MB91V460) or UART4 (MB91460 series flash derivatives) to indicate that the bootloader is ready to accept commands.

There are five different commands supported by the internal bootloader. See the table below for serial protocol of these commands.

Note: Software to transfer data using this protocol is available from Fujitsu.

Table 53.2-2 Boot loader commands

Command	PC to MCU	MCU to PC	Remark
READ	1 (0x01) → 2 (0x02) → Address (4 bytes) Size (2 bytes)	241 (0xF1) 130 (0x82) Binary Dump CheckSum (2 bytes)	Lo, MidLo, MidHi, Hi Lo, Hi Direct read and dump Bootloader sends 16bit checksum
WRITE	1 (0x01) → 3 (0x03) → Address (4 bytes) Size (2 bytes) Binary Dump	241 (0xF1) 131 (0x83) CheckSum (2 bytes)	Lo, MidLo, MidHi, Hi Lo, Hi Receive and store dump in RAM Bootloader sends 16bit checksum
CALL	1 (0x01) → 4 (0x04) → Address (4 bytes)	241 (0xF1) 132 (0x84) Return Parameter	Lo, MidLo, MidHi, Hi Calls specified Address and waits for a return. The returned parameter in R4 will be echoed to the PC
CHECKSUM	1 (0x01) → 5 (0x05) →	241 (0xF1) 133 (0x85) CheckSum (2 bytes)	MCU re-dumps 16bit checksum (Lo, Hi) calculated at last write or read operation
BAUDRATE	1 (0x01) → 6 (0x06) → Baudrate (4 bytes)	241 (0xF1) 134 (0x86)	Lo, MidLo, MidHi, Hi Initializes UART with new baudrate value

53.3. Registers modified by Boot ROM

The Boot ROM initializes the chip and changes the settings of some registers.

53.3.1 Reset Source Register (RSRR)

The RSRR register can only be read once. After reading the RSRR-register, the contents will be present in R4 (C-Compiler convention for parameters) after a branch to application start or the address specified by the Boot-Security-Vector. The RSRR register is also saved in RAM. The RAM address depends on the device. The dependency of the RAM address and the device is listed below.

Table 53.3-1 Reset Source Register (RSRR) data storage addresses

Device	RAM Address
MB91V460A	0x20500
MB91FV460B	programmable ¹
MB91F467Dx	0x28500
MB91F469Gx	0x28500
MB91F463Nx	n.a. ²
MB91F467Rx	n.a. ³
all other MB91460 series devices	0x2E500

1. MB91FV460B contains a programmable flash memory to emulate various device boot ROMs.
2. Device has a special Boot ROM. For more information, please contact Fujitsu.
3. Device has a special Boot ROM. For more information, please contact Fujitsu.

53.3.2 Evaluation Chip MB91V460A

Register	Value	Address	Description
TBCR	0x03	0x482	Time-Base Counter/Sync-RST Register
RSRR	0x00	0x480	Reset Source Register (visible in R4) ¹
TBR	0x0FFC00	-	Table Base Register
SP*	0x0203F8	-	Stack Pointer
PFR21*	0x07	0xD95	Port Function Register Port 21 (UART0)
SMR00*	0x05	0x041	UART0 Mode Register
SCR00*	0x17	0x040	UART0 Control Register
SSR00*		0x042	UART0 Serial Status Register
RDR00*		0x043	UART0 Reception Data Register
TRD00*		0x043	UART0 Transmission Data Register
BGR00*	0x0CF	0x080	UART0 Baud Rate/Reload Counter Register
TMCSR0*	0x0003	0x1B6	Reload Timer 0 Control Status Register
TMRLR0*	0x03E8	0x1B0	Reload Timer 0 Reload Register
TMR0*		0x1B2	Reload Timer 0 Timer Register

1. See 53.3.1 Reset Source Register (RSRR) above

* These registers will only be modified if a check for valid boot condition is performed

53.3.3 Flash devices of MB91460 series

Register	Value	Address	Description
TBCR	0x03	0x482	Time-Base Counter/Sync-RST Register
RSRR	0x00	0x480	Reset Source Register (visible in R4) ¹
TBR	0x0FFC00	-	Table Base Register
SP*		-	Stack Pointer (depends on RAM size)

1. See 53.3.1 Reset Source Register (RSRR) above

* These registers will only be modified if a check for valid boot condition is performed

53.4. Flash Access Mode Switching

The Boot ROM contains a software routine to switch the flash access mode (16 bit / 32 bit / 64 bit).

See chapter [54.6. Flash Access Mode Switching \(Page No.1183\)](#).

53.5. Bootloader Update Strategy

Some applications require the possibility of software updates in the final product without great effort. To be able to program an updated application to the microcontroller, all connected busses (I2C, CAN, LIN, K-Line etc.) can be used, so that the microcontroller can remain 'embedded' in a bigger system.

But therefore a user bootloader has to be integrated to the application software, which handles the startup of the application, the requests for new programming and the application update itself.

This user bootloader should be executed after every reset command, so the Boot Security Vector should point to the starting address of this bootloader. The request for re-programming the application has to be done by a defined communication protocol.

If the Boot Security Vector feature is used, there is also no possibility to enter the internal bootloader of the microcontroller. So the code is secured from read-out or manipulation.

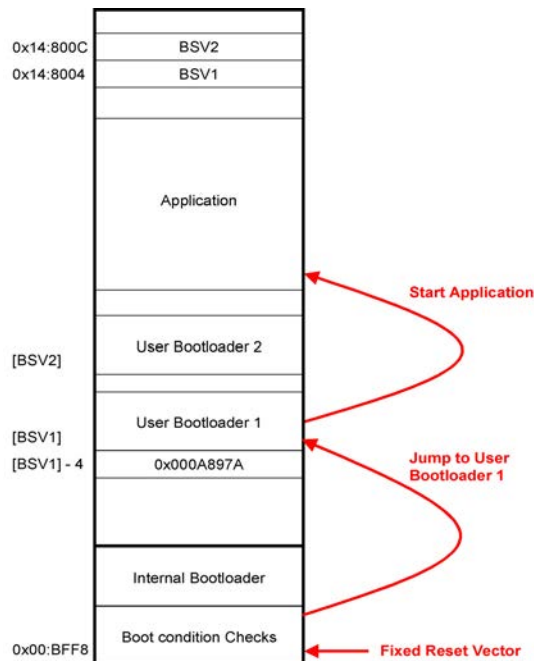
If there is a request for updating an application, the user bootloader causes the erase of flash sectors where the application is located. Flash section, where the Boot Security Vector is located and the section of the bootloader itself may not be erased. After erasing, the bootloader has to handle the programming of the application.

If during this procedure problems like reset or power-down occur, the programming can be started again without problems because user bootloader and Boot Security Vector are not changed.

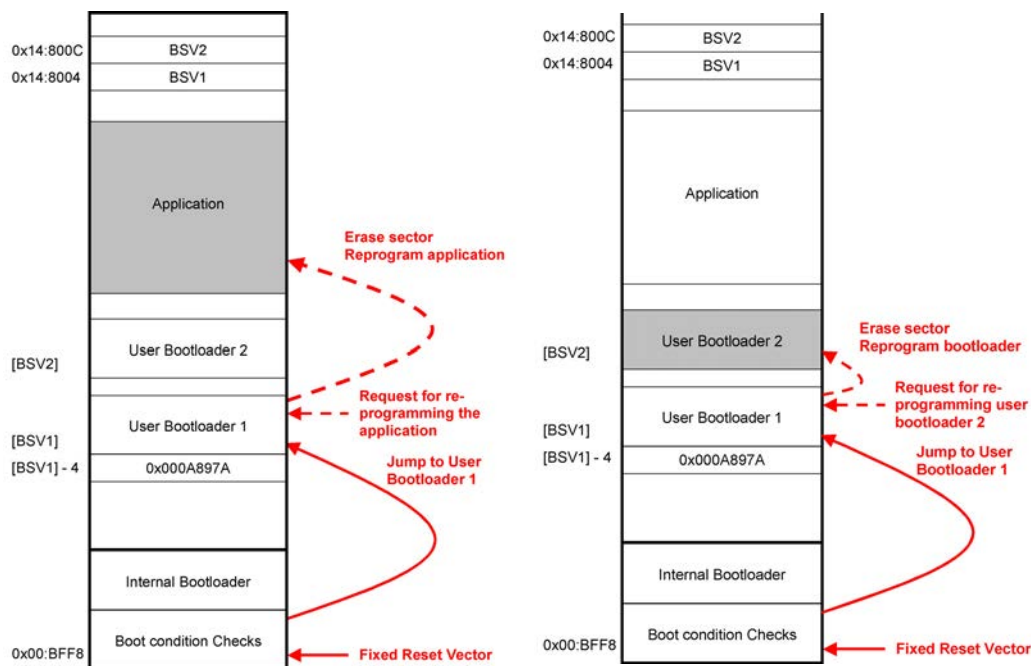
This system brings along some problems if the user bootloader has to be updated. Therefore also the user bootloader sector has to be erased and re-programmed. If there is a reset or power-down before ending the programming, there is no valid bootloader any more and there will be no access to the microcontroller via the above mentioned busses.

All MB91460 series flash devices therefore offer the possibility of a safe bootloader update by use of two Boot Security Vectors and a Magic Number.

If user bootloader 1 is valid, set the Magic Number, which has to be located on the four bytes before the address defined by BSV1, to 0x000A897A. So after every reset this bootloader is executed. If there is no request for a re-programming, the application is started normally.

Figure 53.5-1 Normal start of the application

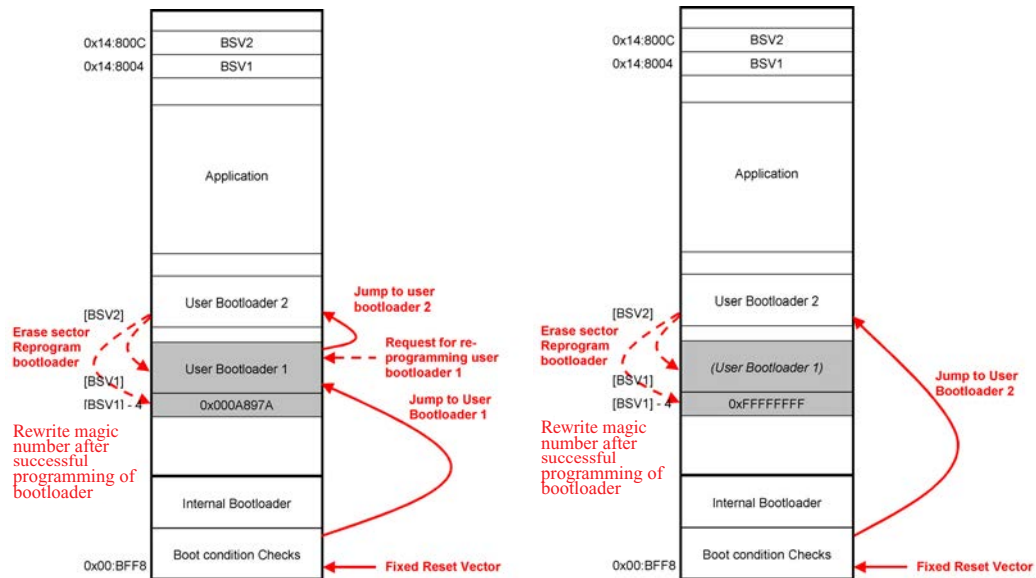
By use of this bootloader, the application as well as user bootloader 2 can be re-programmed:

Figure 53.5-2 The application as well as user bootloader 2 can be re-programmed

If the user bootloader 1 should be re-programmed, then functions in user bootloader 2 have to be called to erase the sector where user bootloader 1 and the Magic Number are located. After erasing, program the new user bootloader to the section. As last step, the Magic Number should be set to 0x000A897A.

This procedure guarantees that there is started always a valid bootloader. If after erasing or during programming user bootloader 1 a reset or a power-down occurs, user bootloader 2 is started and the programming can be resumed.

Figure 53.5-3 Reset during Programming of bootloader 1



53.6. Chip-ID

The chip-IDs of different devices are tabulated below.

Table 53.6-1 List of devices with Chip-ID at location 0xBFF0

DEVICE	Chip-ID location: 0xBFF0	Datecodes
MB91461	0x00000100	
MB91F463NA	0x4630100	
MB91F463NB	0x4630200	
MB91F467RA	0x00000100	
MB91F467RB	0x00000200	
MB91F467RC	0x00000200	

Table 53.6-2 List of Devices with Chip-ID at location 0xBFF4

DEVICE	Chip-ID location: 0xBFF4	Datecodes
MB91V460A	0x020DCC01	
MB91F464AA	Rev. 1 0x0205D048 Rev. 2 0x0205D049 Rev. 3 0x0205D04A	0644-K00, K01, K02
MB91F464AB	0x0205D04B	
MB91F465BB	0x43000040	
MB91F467BA	Rev. 1 0x43000008 Rev. 2 0x43000041	0646-Z00
MB91F467BB	0x43000062	
MB91F467CA	0x43000007	
MB91F467CB	0x43000065	
MB91F465DA	Rev. 1 0x43000005 Rev. 2 0x43000063	
MB91F467DA	Rev. 1 0x0205D329 Rev. 2 0x0205D32A Rev. 3 0x0205D32B Rev. 4 0x0205D32C	0616-Z11 0642-Z07
MB91F467DB	0x0205D32D	
MB91F467EA	0x430000CB	
MB91F469GA	Rev. 1 0x0205D558 Rev. 2 0x0205D558 Rev. 3 0x0205D55A	0643-Z03
MB91F469GB	0x0205D55B	
MB91F465KA	Rev. 1 0x0205D050 Rev. 2 0x0205D051 Rev. 3 0x0205D052	
MB91F465KB	0x0205D053	
MB91F467MA	0x43000021	
MB91F465XA	Rev. 1 0x43000006 Rev. 2 0x43000064	

MB91460 Series

Chapter 54 Flash Memory

This chapter describes the use of the built-in flash memory.

54.1. Overview

The MB91F46x devices as well as MB91FV460B have built-in Flash memory with a variety of capacities, capable of batch-erasing all sectors or erasing on the sector level via single +3.0V-5.5V power supply, and writing by the FR-CPU at the half-word (16-bit) and word (32-bit) level.

54.2. Features

- Power: Single +3.0-5.5V supply
- Basic specification: Same as Spansion MBM29LV400TC (except size and part of sector configuration)
- Faster device operation by enabling commands/data reads at D-word (64-bit) level (not supported by all devices, see [Table 54.6-1](#) or appropriate datasheet)
- External writers: Interface for Parallel Flash Programmer available.
- Operation modes:
 - (1) 64-bit CPU mode (not supported by all devices, see [Table 54.6-1](#) or the datasheets of the devices)
CPU reads and executes programs in word (32-bit) length units.
Flash writing is not possible.
Actual Flash Memory access is performed in d-word (64-bit) length units.
 - (2) 32-bit CPU mode
CPU reads, writes and executes programs in word (32-bit) length units.
Actual Flash Memory access is performed in word (32-bit) length units.
 - (3) 16-bit CPU mode:
CPU reads and writes in half-word (16-bit) length units.
Program execution from the Flash is not possible.
Actual Flash Memory access is performed in word (16-bit) length units.
 - (4) Flash memory mode (parallel programmer mode, external access to Flash memory enabled)
- Features (through combination of Flash memory macro and FR-CPU interface circuit):
 - Functions as CPU program/data storage memory.
 - Enables access to 32-bit bus width.
 - Enables read/write/erase by CPU (auto program algorithm¹).
 - Functions equivalent to Spansion MBM29LV400TC² stand-alone Flash-memory product.
 - Enables read/write/erase by parallel Flash programmer (auto program algorithm¹).

1. Auto program algorithm = Embedded Algorithm TM

2. The Flash memory manufacturer code and device code are not available. The corresponding read commands specified for MBM29LV200TC are not supported.

54.3. Configuration

Figure 54.3-1 Block Diagram (32bit/64bit flash) internal access

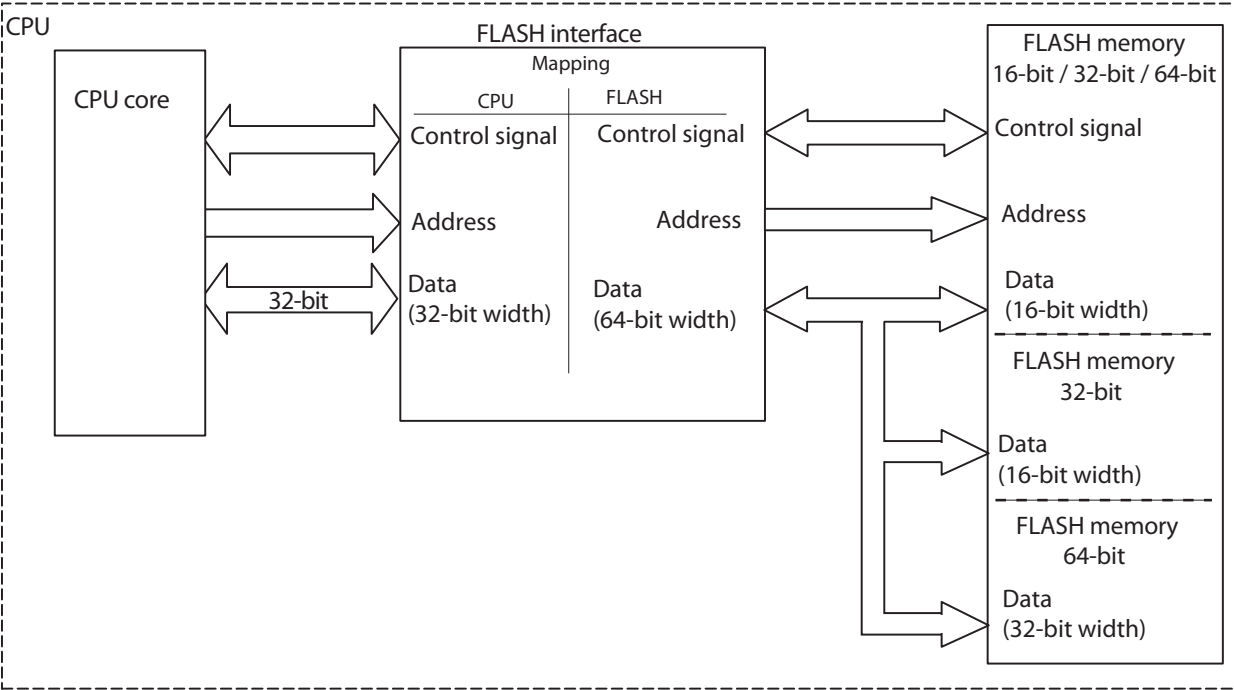
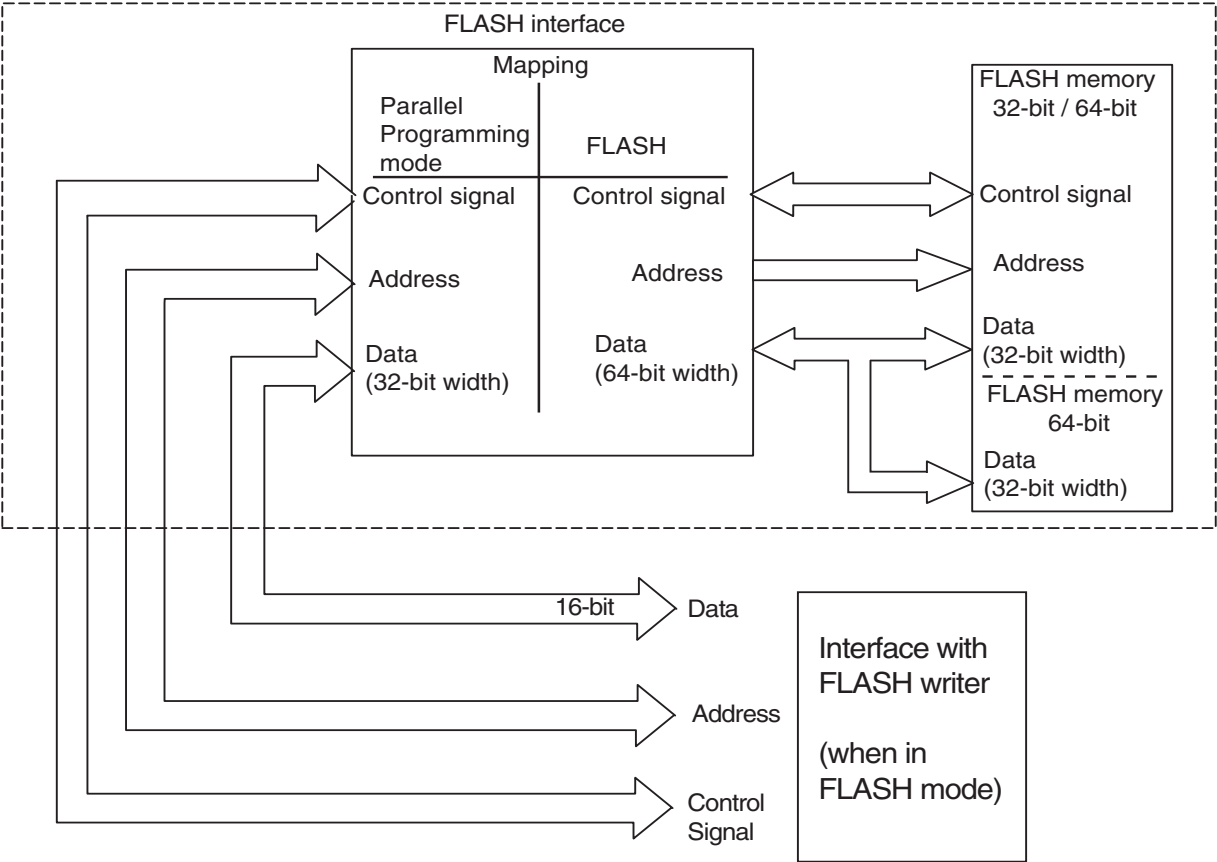


Figure 54.3-2 Block Diagram (32bit/64bit flash) external access



MB91460 Series

54.3.1 Flash Memory Sector Organisation

The following figures show the sector organisation of the 1088 KByte flash macro, as used on MB91F467Dx, in CPU access and in parallel programming mode. For other devices, please refer to the chapter EMBEDDED PROGRAM/DATA MEMORY (FLASH) in the datasheets.

Figure 54.3-3 Flash sector organisation in CPU access

Address										
0014:FFFFh 0014:C000h	SA6 (8KB)				SA7 (8KB)				ROMS7	
0014:BFFFh 0014:8000h	SA4 (8KB)				SA5 (8KB)					
0014:7FFFh 0014:4000h	SA2 (8KB)				SA3 (8KB)					
0014:3FFFh 0014:0000h	SA0 (8KB)				SA1 (8KB)					
0013:FFFFh 0012:0000h	SA22 (64KB)				SA23 (64KB)				ROMS6	
0011:FFFFh 0010:0000h	SA20 (64KB)				SA21 (64KB)					
000F:FFFFh 000E:0000h	SA18 (64KB)				SA19 (64KB)				ROMS5	
000D:FFFFh 000C:0000h	SA16 (64KB)				SA17 (64KB)				ROMS4	
000B:FFFFh 000A:0000h	SA14 (64KB)				SA15 (64KB)				ROMS3	
0009:FFFFh 0008:0000h	SA12 (64KB)				SA13 (64KB)				ROMS2	
0007:FFFFh 0006:0000h	SA10 (64KB)				SA11 (64KB)				ROMS1	
0005:FFFFh 0004:0000h	SA8 (64KB)				SA9 (64KB)				ROMS0	
	addr+0	addr+1	addr+2	addr+3	addr+4	addr+5	addr+6	addr+7		
16bit read/write	dat[31:16]		dat[15:0]		dat[31:16]		dat[15:0]			
32bit read/write	dat[31:0]				dat[31:0]					
64bit read	dat[63:0]									

Note: ROMS0 to ROMS7 are the appropriate bits of the ROMS register.

Figure 54.3-4 Flash sector organisation in parallel programming mode

FA[21:0]	
003F:FFFFh 003F:0000h	SA23 (64KB)
003E:FFFFh 003E:0000h	SA22 (64KB)
003D:FFFFh 003D:0000h	SA21 (64KB)
003C:FFFFh 003C:0000h	SA20 (64KB)
003B:FFFFh 003B:0000h	SA19 (64KB)
003A:FFFFh 003A:0000h	SA18 (64KB)
0039:FFFFh 0039:0000h	SA17 (64KB)
0038:FFFFh 0038:0000h	SA16 (64KB)
0037:FFFFh 0037:0000h	SA15 (64KB)
0036:FFFFh 0036:0000h	SA14 (64KB)
0035:FFFFh 0035:0000h	SA13 (64KB)
0034:FFFFh 0034:0000h	SA12 (64KB)
0033:FFFFh 0033:0000h	SA11 (64KB)
0032:FFFFh 0032:0000h	SA10 (64KB)
0031:FFFFh 0031:0000h	SA9 (64KB)
0030:FFFFh 0030:0000h	SA8 (64KB)
002F:FFFFh 002F:E000h	SA7 (8KB)
002F:DFFFh 002F:C000h	SA6 (8KB)
002F:BFFFh 002F:A000h	SA5 (8KB)
002F:9FFFh 002F:8000h	SA4 (8KB)
002F:7FFFh 002F:6000h	SA3 (8KB)
002F:5FFFh 002F:4000h	SA2 (8KB)
002F:3FFFh 002F:2000h	SA1 (8KB)
002F:1FFFh 002F:0000h	SA0 (8KB)
	FA[1:0]=00FA[1:0]=10
16bit write mode	DQ[15:0]DQ[15:0]

Note: Always keep FA[0] = 0 and FA[21] = 1 in parallel programming mode.

MB91460 Series

54.3.2 Address conversion from CPU Mode to Flash Programming Mode

Flash memory's address mapping is different depending on whether it is being accessed from the FR-CPU or parallel Flash programmer.

The following equations are an **example** to calculate a Flash programming mode address (FA) from a CPU mode address (addr) for the 64bit flash type with a memory size of 1088kB (16x64kB + 8x8kB), as used on MB91F467Dx:

■ Even 8KB sectors

SA0, SA2, SA4, SA6 (14:0000h ≤ addr ≤ 14:FFFFh; addr[2]=0):

$$FA = \text{addr} - \text{addr}\%00:4000\text{h} + (\text{addr}\%00:4000\text{h})/2 - (\text{addr}/2)\%4 + \text{addr}\%4 - 05:0000\text{h}$$

■ Odd 8KB sectors

SA1, SA3, SA5, SA7 (14:0000h ≤ addr ≤ 14:FFFFh; addr[2]=1):

$$FA = \text{addr} - \text{addr}\%00:4000\text{h} + (\text{addr}\%00:4000\text{h})/2 - (\text{addr}/2)\%4 + \text{addr}\%4 - 04:E000\text{h}$$

■ Even 64KB sectors

SA8, SA10, SA12, SA14, SA16, SA18, SA20, SA22 (04:0000h ≤ addr ≤ 13:FFFFh; addr[2]=0):

$$FA = \text{addr} - \text{addr}\%02:0000\text{h} + (\text{addr}\%02:0000\text{h})/2 - (\text{addr}/2)\%4 + \text{addr}\%4 + 0C:0000\text{h}$$

■ Odd 64KB sectors

SA9, SA11, SA13, SA15, SA17, SA19, SA21, SA23 (04:0000h ≤ addr ≤ 13:FFFFh; addr[2]=1):

$$FA = \text{addr} - \text{addr}\%02:0000\text{h} + (\text{addr}\%02:0000\text{h})/2 - (\text{addr}/2)\%4 + \text{addr}\%4 + 0D:0000\text{h}$$

For other flash memory sizes or types please see the related datasheets.

54.4. Registers

Table 54.4-1 gives an overview about the registers related to the Main Flash. For the detailed description, please refer to section 11.7. [Registers \(Page No.264\)](#) in [Chapter 11 Memory Controller, Flash and F-Cache](#).

Table 54.4-1 Main Flash Register Overview

Address	7	6	5	4	3	2	1	0	Register
07000 _H	ASYNC	FIXE	(BIRE)	RDYEG	RDY	RDYI	RW16	—	FMCS
07001 _H	—	—	—	—	LOCK	(PHASE)	PF2I	RD64	FMCR
07004 _H	WTP1	WTP0	WEXH1	WEXH0	WTC3	WTC2	WTC1	WTC0	FMWT
07005 _H	(FRAM)	ATD2	ATD1	ATD0	EQ3	EQ2	EQ1	EQ0	
07006 _H	—	ALEH2	ALEH1	ALEH0					FMWT2

54.5. Access Modes

This section describes the Flash memory access modes.

54.5.1 Access from the FR-CPU

The following three types of access mode are available:

■ 64-bit CPU mode (read/execute)

This mode is not supported by all devices, see [Table 54.6-1](#) or the appropriate datasheet.

This mode does not allow data erase/write. Data can only be accessed by the CPU in lengths of words (32 bits) but is actually read from the Flash Memory in lengths of d-words (64 bits).

Programs can be executed in Flash memory while this mode is enabled.

- Specifying the mode

Use the Flash setting procedure located in the Boot-ROM to set this mode. Refer to [54.6. Flash Access Mode Switching \(Page No.1183\)](#).

- Description of operation

When reading or executing code from the Flash memory area, data is read by the CPU in word (32-bit) length units but is actually read from memory in d-word (64-bit) length units.

Running Auto Program Algorithms is not possible.

■ 32-bit CPU mode (read/write/execute)

Write is not supported by all devices, see [Table 54.6-1](#) or the appropriate datasheet.

This mode allows data erase/write. Data can only be accessed in lengths of words (32 bits).

Programs cannot be executed in Flash memory while the Flash is being written/erased.

- Specifying the mode

Use the Flash setting procedure located in the Boot-ROM to set this mode. Refer to [54.6. Flash Access Mode Switching \(Page No.1183\)](#).

Flash memory always goes to 32-bit mode after a reset is cleared, when the CPU is running.

- Description of operation

When reading or executing code from the Flash memory area, the datawidth is 32 bit.

Auto Algorithms can be run by writing commands to Flash memory. It is possible to erase/write to Flash memory by running an Auto Program Algorithm. See “[54.7. Auto Program Algorithms \(Page No.1185\)](#)” for details about Auto Program Algorithms.

■ 16-bit CPU mode (read/write)

This mode allows data erase/write. Data can only be accessed in lengths of half-words (16 bits).

MB91460 Series

Programs cannot be executed in Flash memory while this mode is enabled.

- Specifying the mode

Use the Flash setting procedure located in the Boot-ROM to set this mode. Refer to [54.6. Flash Access Mode Switching \(Page No.1183\)](#).

- Description of operation

When reading from the Flash memory area, data is read from memory in half-word (16-bit) length units.

Auto Program Algorithms can be run by writing commands to Flash memory. It is possible to erase/write the Flash memory by running an Auto Program Algorithm. See “[54.7. Auto Program Algorithms \(Page No.1185\)](#)” for details about Auto Program Algorithms.

54.5.2 Flash Memory Mode (Parallel Programmer Mode)

Resetting after setting the MD_2, MD_1 and MD_0 pins to “1”, “1” and “1” will halt the MCU functions. At this time, the Flash memory's interface circuit functions are enabled for direct control of the Flash memory unit from external pins, by directly linking some of the signals to the Flash memory unit's control signal. In this mode, the Flash memory appears to the external pins as a stand-alone unit. This mode is generally set when writing/erasing using the parallel Flash programmer. In this mode, all operations of the Flash memory's Auto Program Algorithms are available similar to the CPU access mode.

For correspondence between the MCU pins and Flash memory's interface circuit in the flash parallel programming mode, please check the related datasheets of each product.

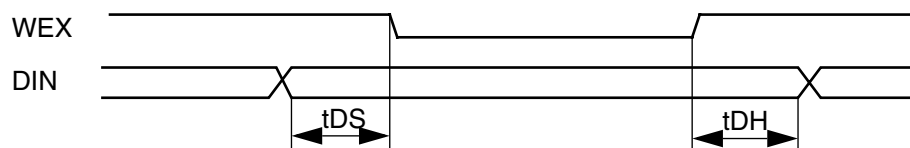
● Poweron Sequence in parallel programming mode

The flash memory can be accessed in programming mode after a certain wait time, which is needed for Security Vector fetch:

- Minimum wait time after VDD5/VDD5R power on: 2.76 ms
- Minimum wait time after INITX rising: 1.0 ms

● Timing Requirement of the Flash Security

To avoid unexpected cancellation of flash write sequences by the Flash Security, make sure that the write data does not change during WEX=L. Check that the timing of write data DIN versus WEX is like shown here.



MB91460 Series

54.6. Flash Access Mode Switching

On MB91460 series flash devices, it is possible to switch between different CPU flash access modes. These modes are depending on device type.

Table 54.6-1 MB91460 series CPU flash access modes

Device	Flash KBytes	16bit access	32bit access	64bit access	Boot ROM address ¹
MB91V460A	—	—	—	—	—
MB91F463CA	288	read/write	read	—	0BF60 _H
MB91F463Nx	288	read/write	read	—	—
MB91F464Ax	416	read/write	read	—	0BF60 _H
MB91F464BB					
MB91F464Hx					
MB91F465BB	544	read/write	read	—	0BF60 _H
MB91F465CA					
MB91F465DA					
MB91F465Kx					
MB91F465PA					
MB91F465XA					
MB91F466Bx	832	read/write	read/write	read	0BF60 _H
MB91F466HA					
MB91F467Bx	1088	read/write	read/write	read	0BF60 _H
MB91F467Cx					
MB91F467Dx					
MB91F467EA					
MB91F467MA					
MB91F467PA					
MB91F467Rx	1088	read/write	read/write	read	—
MB91F467Sx	1088	read/write	read/write	read	0BF60 _H
MB91F467TA					
MB91F469Gx	2112	read/write	read/write	read	0BF60 _H
MB91F469QA					
MB91F469TA					
MB91FV460B	2112	read/write	read/write	read	programmable ²

1. Address of the access mode switching routine in Boot ROM
2. MB91FV460B contains a programmable flash memory to emulate various device boot ROMs

The 16bit mode is intended basically for flash programming, so program execution from flash is prohibited anyway. The 32bit and 64bit modes, however, are intended for program execution from flash. Since switching between these modes directly from an application located in the flash is prohibited, it is necessary to locate the switching routine either in RAM, or use the available routine in the boot ROM at the addresses listed in [Table](#)

54.6-1 “MB91460 series CPU flash access modes,” on page 1183 above.

To call this function two parameters are expected in CPU registers R4 and R5:

R4: 2bits (LSB of register) for setting the flash access mode

b'00:	32bit read/write mode
b'01:	16bit read/write mode
b'1x:	64bit read only mode

R5: 16bits for setting wait time

The wait time should be in the range of some 100ns after switching the flash access mode for stabilization of the flash memory. The wait function is a simple delay loop that needs about 3 CPU cycles for one run. The value in R5 represents the number of runs through this delay loop.

After leaving this delay loop, the switching function will return to the application.

MB91460 Series

54.7. Auto Program Algorithms

Write and erase to Flash memory are performed by launching the Flash memory's own Auto Program Algorithms. The following commands are available: Read/Reset, Write, Chip Erase and Sector Erase. The erase suspend and restart function is available during Sector erase.

Please disable the Flash Cache before launching an Auto Program Algorithm and flush the F-Cache after the programming is finished.

54.7.1 Command Sequences

Auto Program Algorithms are launched by writing one to six bytes or half words (16 bits) to the Flash memory in succession according to [Table 54.7-1 Flash command sequences](#). This table is valid for the Program/Data Flash (Main Flash) in CPU mode and parallel programmer mode (Flash mode).

Writing the command sequences is possible in byte, halfword or word¹ mode. The table shows halfword data ("F0F0_H") because it simplifies the calculation of the addresses.

The data width used for the 4th bus write cycle of the write command (program address and program data) determines the write mode of the Flash (byte, halfword or word¹).

Writing an illegal address or data, or writing them in the incorrect order, will reset the Flash memory to read mode.

Table 54.7-1 Flash command sequences

Command Sequence	Bus Cycles	1st Bus Write cycle		2nd Bus Write cycle		3rd Bus Write cycle		4th Bus Read/write cycle		5th Bus Write cycle		6th Bus Write cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/reset	1	*1	F0F0 _H	-	-	-	-	-	-	-	-	-	-
Read/reset	4	*2	AAAA _H	*3	5555 _H	*2	F0F0 _H	RA	RD	-	-	-	-
Write	4	*2	AAAA _H	*3	5555 _H	*2	A0A0 _H	PA	PD	-	-	-	-
Chip erase	6	*2	AAAA _H	*3	5555 _H	*2	8080 _H	*2	AAAA _H	*3	5555 _H	*2	1010 _H
Sector erase	6	*2	AAAA _H	*3	5555 _H	*2	8080 _H	*2	AAAA _H	*3	5555 _H	SA	3030 _H
Sector erase suspend	1	*1	B0B0 _H	-	-	-	-	-	-	-	-	-	-
Sector erase resume	1	*1	3030 _H	-	-	-	-	-	-	-	-	-	-
Continuous mode set	3	*2	AAAA _H	*3	5555 _H	*2	2020 _H	-	-	-	-	-	-
Continuous writing	2	*2	A0A0 _H	PA	PD	-	-	-	-	-	-	-	-
Continuous mode reset	2	*1	9090 _H	*1	F0 _H or 00 _H	-	-	-	-	-	-	-	-

RA: Read address

PA: Write address (program address)

RD: Read data (16-bit / 32-bit)

PD: Program data (16-bit / 32-bit)

SA: Sector address, specifies an address within the sector

*1: any valid address pointing into a sector that is not write protected

*2: In CPU mode: Address= (adr & FFFC000_H) + 1553_H

In Flash mode: Address= (adr & FFFF000_H) + AAB_H

"adr" is the target address

*3: In CPU mode: Address= (adr & FFFC000_H) + 0AAB_H

In Flash mode: Address= (adr & FFFF000_H) + 557_H

or an address within the target sector

About valid flash sectors, please refer to the datasheets:

- section "Flash access in CPU mode" to find flash addresses in CPU mode
- section "Flash configuration in parallel Flash programming mode" for addresses in flash mode

Examples of the 1088KByte flash, used on MB91F467Dx, can be found in [Figure 54.3-3 Flash sector or-](#)

1. word access in command sequences is possible only on devices with 32-bit write access

ganisation in CPU access (Page No.1177) and Figure 54.3-4 Flash sector organisation in parallel programming mode (Page No.1178).

■ Auto Program Algorithm Execution Status

If an Auto Program Algorithm is started in CPU mode, it is possible to learn the operational state of the Auto Program Algorithm via the internal ready signal (RDY). The level of the ready signal can be read from the “RDY” bit of the FLASH Memory Control Status Register.

While the “RDY” bit is set to “0”, data read is the hardware sequence flag indicating Flash memory status (see [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#)).

54.7.2 Auto Program Algorithm Commands

■ Read/reset command

Issue a Read/reset command sequence to recover to read mode after a timing limit has been exceeded. Data is read from Flash memory via the read cycle. Flash memory stays in a read state until another command is input.

When powered up, Flash memory is automatically set to read/reset. In this case, commands are not required for data reading.

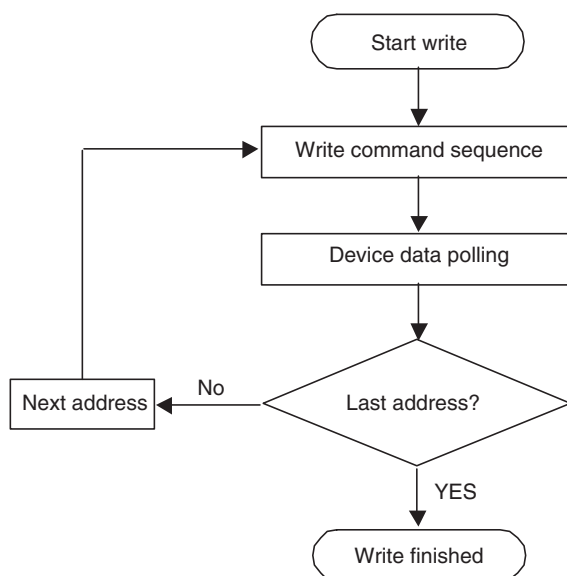
■ Program (write)

In CPU programming mode, writes are performed in basic units of half words. Writes are performed in 4 bus operations. The command sequence consists of two “unlock” cycles, followed by a write setup command and write data cycle. Then, in the final write cycle, writing to memory starts.

After the auto write algorithm command sequence is executed, Flash memory no longer requires external control. Flash memory generates appropriate write pulses that it has automatically created internally, and validates the margins of written cells. Auto write operation ends when the bit 7 data matches the data written to this bit via data polling (see [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#)). Flash memory then returns to read mode, and no longer accepts write addresses. As a result, at this time Flash memory requests the next valid address. Thus, data polling indicates that writing is ongoing.

During writing, all commands written to Flash memory are ignored. If a hardware reset is started during writing, the data in addresses that have been written is not guaranteed. Data can be written to addresses in any order, and may also cross sector boundaries. Writing cannot return data “0” to data “1”. If data “1” is written to data “0”, then either the data polling algorithm will determine that the device is bad, or it will appear that data “1” has been written, but in reset/read mode, when the data is read, it will still be read as “0”. Only erase operation can change “0” data to “1” data.

Figure 54.7-1 The Writing Sequence Using Write Commands



■ Chip erase

Chip erase (erase all sectors at once) is performed via six accesses. First, there are two “unlock” cycles, after which a setup command is written. This is then followed by two more “unlock” commands before the chip erase command.

The user does not have to write to Flash memory before a chip erase can be performed. While the auto erase algorithm is executing, Flash memory automatically validates each cell before erasing it by writing a “0” pattern (preprogram). During this operation, the Flash memory does not require external control.

Auto erase begins with a write during the command sequence, and ends when “1” is written to bit 7, at which point the Flash memory returns to read mode. The time for chip erase is equal to [sector erase time] x [number of sectors] + [chip write time (preprogram)].

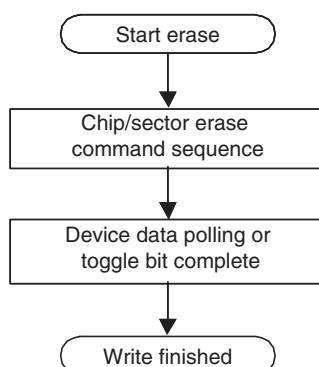
The figure below shows the chip erase sequence using the chip erase command.

■ Sector erase

Sector erase is performed via six accesses. There are two “unlock” cycles, after which a “setup” command is written, followed by another two “unlock” cycles. On the sixth cycle, a sector erase command is input, starting the sector erase. From the time that the last sector-erase command is written until the timeout of 50 μ s, the next sector-erase command will be accepted.

It is possible to submit multiple sector erases simultaneously by writing the six bus cycles described above. This sequence is performed by writing the addresses of sectors to erase in succession after the sector-erase command (30_H). After a timeout of 50 μ s since the last sector-erase command was written, sector erase begins. In other words, to erase multiple sectors simultaneously, each sector must be entered within 50 μ s of the other, after which commands may no longer be accepted. It is possible to monitor whether successive sector-erase commands are valid via bit 3 (see [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#)). After finishing, Flash memory returns to read mode. Other commands are ignored. Data polling works on any address in an erased sector. The time for multiple-sector erase is equal to ([sector erase time] + [sector write time (preprogram)]) x [number of sectors erased].

Figure 54.7-2 Chip Erase Sequence Using the Chip erase Command



■ Erase suspend

The erase suspend command allows the user to pause the Flash memory's Auto Program Algorithm during sector erase, and read data from/write data to sectors not being erased. This command is only valid during sector erase. It is ignored during chip erase and write operations. The erase suspend command (B0_H) is only valid during sector-erase operation, including the timeout period after a sector-erase command (30_H). Entering this command during the timeout period immediately ends the timeout, and interrupts the erase operation. When the erase resume command is written, the erase operation resumes. Any address can be used for erase suspend and erase resume command input.

If a erase suspend command is input during sector-erase operation, it will take up to 20 μ s for the Flash memory to halt the erase operation. When the Flash memory goes into erase suspend mode, it outputs ready/busy and bit 7 outputs "1", and bit-6 toggling is halted. It is possible to confirm whether the erase operation has halted, by entering the address of an erased sector, and monitoring the values read from bits 6 and 7. Additionally, writes of erase-suspend commands are ignored. When erasing is halted, the Flash memory goes into erase-suspend read mode. In this mode, data reads from sectors where erase has not been paused are enabled, but for other sectors, it is the same as standard reading. While in erase-suspend read mode, when data is read sequentially from an erase-suspended sector, bit 2 is toggled (see [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#) for details).

After entering erase-suspend read mode, the user can write to the Flash memory by writing a write command sequence. This mode is called "erase-suspend write mode". In this mode, data writes to sectors where erase has not been paused are enabled, but for other sectors, it is the same as normal byte writing. While in erase-suspend write mode, when data is read sequentially from an erase-suspended sector, bit 2 is toggled. This mode can be detected via the erase-suspend bit (bit 6).

A word of caution is required for using this mode: Although bit 6 can be read from any address, bit 7 must be read from a write address. To resume sector erase, a resume command must be entered (30_H). At this point, further resume commands will be ignored. Conversely, it is possible to enter a erase-suspend command after the Flash memory resumes erasing.

Table 54.7-4 Hardware sequence flag functions

Status		Accessed address	DQ7 DPOLL	DQ6 TOGGLE	DQ5 TLOVER	DQ4 ¹	DQ3 SETIMR	DQ2 TOGGL2
Normal operation	Read/Reset	Any address	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	Write (embedded program algorithm running)	Address being programmed	DATA:7	Toggle	0	0	0	1
	Sector erase wait (Flash waits for inputting further sector erase requests)	Sectors which are already specified for erase	1	Toggle	0	0	0	Toggle
		Other sectors						1
	Chip/sector erase (embedded erase algorithm running)	Sectors which are specified for erase (all sectors in case of Chip erase)	0	Toggle	0	1: busy to suspend 0: ready to suspend	1	Toggle
		Other sectors						1
	Sector erase suspended	Sectors which are specified for erase	1	1	0	0	0	Toggle
		Other sectors	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	Writing of another sector while Flash is in erase suspended state	Address being programmed	DATA:7	Toggle	0	0	0	1
Timing limit exceeded	Chip/sector erase finished (Read/Reset state)	Sector which was erased	DATA:7 = 1	DATA:6 = 1	DATA:5 = 1	DATA:4 = 1	DATA:3 = 1	DATA:2 = 1
	Write	Address being programmed	DATA:7	Toggle	1	0	0	1
		Sector which caused the Timeout during erase	0	Toggle	1	undefined	1	Toggle
	Chip/sector erase	Other sectors						1

1. The DQ4 flag is available only for the Data Flash.

54.7.4 Data Polling Flag (DPOLL, DQ7)

The data polling flag (DPOLL, DQ7) indicates if the automatic algorithm (write or erase) is being executed or has terminated. The function of the DQ7 flag in each flash status is listed in [Table 54.7-4 Hardware sequence flag functions \(Page No.1190\)](#) and explained here.

- **Write**

A read access during execution of the automatic write algorithm causes the flash memory to output the inverted value of bit 7 of the program data ($\overline{\text{DATA}}[7]$) regardless which Flash address is being read.

A read access after termination of the automatic write algorithm is handled as a regular Flash read access and returns bit 7 (DATA:7) of the currently addressed Flash cell. Hence for correctly identifying the termination of a write command, polling should always be performed from the memory location which is being programmed.

- **Word and Byte writing**

A word write command writes data to the high-order byte (DATA[15:8]) and low-order byte (DATA[7:0]). DQ7 shows the inverted value of DATA[7].

A byte write command to an even address writes data to the low-order byte at DATA[7:0] only. DQ7 shows the inverted value of DATA[7].

A byte write command to an odd address writes data to the high-order byte at DATA[15:8] only. In this case DQ7 shows the inverted value of DATA[15].

- **Chip/sector erase**

A read access during execution of the automatic erase algorithm causes the flash memory to output a 0 regardless which Flash address is being read. During the sector erase wait time however, 1 is output.

A read access after termination of the automatic erase algorithm is handled as a regular Flash read access and returns the data of the currently addressed Flash cell. Accessing an erased cell returns 1.

- **Sector erase suspend**

A read access in sector erase suspend state causes the flash memory to output DQ7 = 1 if the address belongs to the sector being erased.

The flash memory outputs bit 7 (DATA: 7) of the addressed memory cell if the address does not belong to the sector being erased.

Referencing this flag together with the toggle bit flag (DQ6) enables a decision to be made on whether the flash memory is in the erase suspended state and which sector is being erased.

Note: When the automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the DQ7 bit can only be identified safely if the corresponding data has been read at least twice.

Note: For the Program/Data Flash (Main Flash), the hardware sequence flags must always be read from an odd address, even when writing a byte to an even address. This means the hardware sequence flags cannot be read from the same address which was written. During programming, DQ7 shows $\overline{\text{DATA}}[15]$. But after termination of the automatic algorithm, the currently addressed Flash cell is read. Depending on the content of this Flash cell, there may be no transition visible on bit 7. Hence the DQ7 bit cannot be used to identify the termination of a byte write to an even address.

54.7.5 Toggle Bit Flag (TOGGLE, DQ6)

The toggle bit flag (DQ6) together with the data polling flag (DQ7) indicates if the automatic algorithm (write or erase) is being executed or has terminated. The function of the DQ6 flag in each flash status is listed in [Table 54.7-4 Hardware sequence flag functions \(Page No.1190\)](#) and explained here.

- Write and chip/sector erase

Successive read accesses during execution of the automatic write or erase algorithm causes the flash memory to toggle the DQ6 flag for every read cycle, regardless which Flash address is being read.

A read access after termination of the automatic algorithm is handled as a regular Flash read access and returns bit 6 (DATA:6) of the currently addressed Flash cell. Accessing an erased cell returns 1.

- Sector erase suspend

A read access in sector erase suspend state causes the flash memory to output DQ6 = 1 if the address belongs to the sector being erased.

The flash memory outputs bit 6 (DATA: 6) of the addressed memory cell if the address does not belong to the sector being erased.

Note: When the automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the toggle bit can only be identified safely if the corresponding data has been read at least three times.

Note: For correctly reading the DQ6 flag of the Program/Data Flash (Main Flash), the following rule must be adhered:

- Make sure at least one other instruction is executed between two read accesses to the hardware sequence flags (at least one NOP).

This rule is not applicable for the Data Flash.

54.7.6 Timing Limit Exceeded Flag (TLOVER, DQ5)

The timing limit exceeded flag (DQ5) indicates that the execution of the automatic algorithm has exceeded the timing (internal pulse count) prescribed in the flash memory. The function of the DQ5 flag in each flash status is listed in [Table 54.7-4 Hardware sequence flag functions \(Page No.1190\)](#) and explained here.

- Write and chip/sector erase

A read access during execution of the automatic write or erase algorithm causes the flash memory to output the status of the DQ5 flag, regardless which Flash address is being read.

A read access after successful termination of the automatic write or erase algorithm is handled as a regular Flash read access and returns bit 5 (DATA:5) of the currently addressed Flash cell.

The DQ5 flag is read as 0 as long as the program or erase time is within the prescribed time (maximum time required for write/erase). After this time has been exceeded, 1 is returned as read value.

An unsuccessful write or erase operation can be determined if DQ5 is 1 while DQ6 and DQ7 shows that the automatic algorithm is still being executed.

For example, writing 1 to a flash memory address where 0 has been written will cause the fail state to occur. In this case, the flash memory will lock and execution of the automatic algorithm will not terminate. As a result, valid data will not be output from the data polling flag (DQ7). In addition, the toggle bit flag (DQ6) will exceed the time limit without stopping the toggle operation and the timing limit exceeded flag (DQ5) will output 1. Note that this state does not indicate that the flash memory is faulty, but has been used incorrectly. When this state occurs, execute the Read/Reset command.

- Sector erase suspend

A read access in sector erase suspend state causes the flash memory to output DQ5 = 0 if the address belongs to the sector being erased.

The flash memory outputs bit 5 (DATA: 5) of the addressed memory cell if the address does not belong to the sector being erased.

Note: When the automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the DQ5 bit can only be identified safely if the corresponding data has been read at least twice.

54.7.7 Sector erase state flag (DQ4)

The DQ4 flag is available only for the Data Flash to improve the function "Suspend Sector Erase".

It is allowed to suspend a sector erase operation more frequently, when the following rules are obeyed:

- During sector erase, the DQ4 flag must be polled with a maximum read period of 20μs.
- If DQ4 is 1 (busy to suspend), polling must continue.
- If DQ4 becomes 0 (ready to suspend), the erase suspend command must be written immediately.
- The maximum length of the "busy to suspend" period is 2.4ms. Within this time, no suspend command should be written to the flash.

Please refer to [54.8.5 Suspending Sector Erase \(Page No.1201\)](#).

54.7.8 Sector Erase Timer Flag (SETIMR, DQ3)

The sector erase timer flag (DQ3) indicates if the execution of the sector erase command has been started of if the sector erase wait period is being applied which allows to submit further sector erase commands.

After submitting a sector erase command sequence, the sector erase wait period is applied. Within this period it is possible to submit further sector erase commands. The DQ3 flag can be used to identify this wait period.

The function of the DQ3 flag in each flash status is listed in [Table 54.7-4 Hardware sequence flag functions \(Page No.1190\)](#) and explained here.

● Sector erase

A read access during execution of the automatic erase algorithm causes the flash memory to output the status of the DQ3 flag, regardless which Flash address is being read. A read access during the sector erase wait period returns 0. After exceeding this wait period, the actual sector erase starts and a read access to the DQ3 flag returns 1. Further sector erase commands will be ignored when DQ3 is 1.

A read access after successful termination of the automatic erase algorithm is handled as a regular Flash read access and returns bit 3 (DATA:3) of the currently addressed Flash cell. Accessing an erased cell returns 1.

See [54.8.4 Erasing Optional Data \(Sector erase\) \(Page No.1199\)](#) for more details about submitting multiple sector erase commands.

The DQ3 flag should be checked before and after submitting further sector erase commands. If DQ3 is read as 1 after submitting the erase command, then this additional sector erase request may not be accepted.

● Sector erase suspend

A read access in sector erase suspend state causes the flash memory to output DQ3 = 1 if the address belongs to the sector being erased.

The flash memory outputs bit 3 (DATA: 3) of the addressed memory cell if the address does not belong to the sector being erased.

Note:When the automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the DQ3 bit can only be identified safely if the corresponding data has been read at least twice.

54.7.9 Toggle Bit-2 Flag (TOGGL2, DQ2)

The toggle bit-2 flag (DQ2) indicates if a sector is in the erase-suspended state. It can also be used to check which sectors are specified for erase.

The DQ2 toggle bit can be used together with the DQ6 toggle bit to determine whether the Flash is in the sector erase suspend state or if the erase algorithm is currently being executed.

The function of the DQ2 flag in each flash status is listed in [Table 54.7-4 Hardware sequence flag functions \(Page No.1190\)](#) and explained here.

- **Sector erase**

Successive read accesses during execution of the automatic sector erase algorithm causes the flash memory to toggle the DQ2 flag for every read cycle if the read address points to a sector which is specified for erase. A read access to another sector returns 1. This can be used to determine if the currently accessed sector belongs to the sectors which are specified for erase. For a chip erase, DQ2 toggles for all sectors.

A read access after termination of the automatic algorithm is handled as a regular Flash read access and returns bit 2 (DATA:2) of the currently addressed Flash cell. Accessing an erased cell returns 1.

- **Sector erase suspend**

Successive read accesses during sector erase suspend also causes the flash memory to toggle the DQ2 flag for every read cycle if the read address points to a sector which is specified for erase. A read access to another sector returns bit 2 (DATA:2) of the currently addressed Flash cell.

Both DQ2 and DQ6 are used for detecting an erase-suspended sector (DQ2 toggles, but DQ6 does not).

Sectors which are not specified for erase can be programmed in erase suspend state. Reading from such a sector during erase-suspend-program mode returns DQ2=1.

- **Timing limit exceeded**

In case of an unsuccessful chip or sector erase operation (Timing limit exceeded, DQ5=1), DQ2 can be used to identify which sector caused the timeout state. DQ2 will only toggle when accessing the sector which caused the timeout. For other sectors, 1 is read.

Note: When the automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the toggle bit can only be identified safely if the corresponding data has been read at least three times from the same location.

Note: For correctly reading the DQ2 flag of the Program/Data Flash (Main Flash), the following rules must be adhered:

- Make sure at least one other instruction is executed between two read accesses to the hardware sequence flags (at least one NOP).
- Do not access flash memory control registers (address 07000_H to 07007_H) between reading the hardware sequence flags.

These rules are not applicable for the Data Flash.

54.8. Detailed Explanation of Writing to and Erasing Flash Memory

This section describes each operation procedure of the flash memory: Read/Reset, Write, Chip Erase, Sector Erase, Sector Erase Suspend and Sector Erase Restart.

By issuing a command sequence (see [Table 54.7-1 “Flash command sequences,” on page 1185](#)) the flash memory executes the automatic algorithm to perform Read/Reset, Write, Chip Erase, Sector Erase, Sector Erase Suspend or Sector Erase Restart operations.

In between write accesses of the command sequence, other bus read/write cycles can be performed as long as no write access to the flash memory other than defined by the command sequence is performed. Even reading the Flash between the write accesses is possible.

Termination of the automatic algorithm can be determined by polling the hardware sequence flags or by polling the RDY flag (FMCS:RDY for Program/Data Flash). At normal termination, the flash memory returns to the read/reset state.

To maintain data consistency it is strongly recommended to disable the Flash Cache while writing to the flash memory and to flush the Flash Cache after completing the auto program algorithm.

Each operation of the flash memory is described in the following chapters:

- Setting the read/reset state
- Writing data by submitting the write command sequence
- Write to the Data Flash by using the Write Command Sequencer
- Erasing all data (chip erase)
- Erasing optional data (sector erase)
- Suspending sector erase
- Restarting sector erase

54.8.1 Setting The Read/Reset State

The read/reset state is the initial state of the flash memory. When the power is turned on and when a command terminates normally, the flash memory is set to the read/reset state. In the read/reset state, any command can be input.

In the Timeout state (DQ5=1), the flash memory can be set to the read/reset state by sending the Read/Reset command in the command sequence table (see [Table 54.7-1 “Flash command sequences,” on page 1185](#)) continuously to the target sector in the flash memory.

The Read/Reset command has two types of command sequences that execute after the first and after the third bus operation. However, there is no essential difference between those two command sequences. The two types of the command are offered for software compatibility.

The Read/Reset command is not required to read data by a regular read. The Read/Reset command is mainly used to initialize the automatic algorithm when a command does not terminate normally.

The Read/Reset command has no effect in normal operation of the write or erase automatic algorithm. It cannot be used to interrupt an ongoing write or erase command execution. Also resetting the Flash from the sector erase suspended state is not possible. The Read/Reset command has no impact on the activation of the fast program mode.

54.8.2 Writing Data by submitting the write command sequence

The data write automatic algorithm of the flash memory can be started by sending the Write command in the command sequence table (see [Table 54.7-1 “Flash command sequences,” on page 1185](#)) continuously to the target sector in the flash memory. When data write to the target address is completed in the fourth command cycle, the automatic algorithm for writing is started.

■ Specifying addresses

Writing to the Program/Data Flash (Main Flash) is possible in byte, halfword or word mode¹.

Writing can be done in any order of addresses. However, the Write command writes only data of one byte or halfword or word for each execution.

■ Notes on writing data

Writing cannot return a data bit in the Flash from 0 to 1. When trying to program a bit to 1 which is already set to 0, the data polling algorithm (DQ7) or toggle operation (DQ6) does not terminate and the timing limit exceeded flag (DQ5) will be set after the prescribed maximum time for writing. A bit programmed to 0 can only be set to 1 by an erase operation.

All commands are ignored during execution of the automatic write algorithm. If a CPU reset (Power on reset, external reset, software reset or watchdog reset) is asserted during writing, the data of the written addresses will be unpredictable. A write access to a write protected sector also asserts the flash hardware resets and cancels an ongoing write operation.

■ Continuous write mode

The continuous write mode (also called “fast write mode”) can be used for programming multiple data to the Flash. After writing the “Set to continuous mode” command to the Flash, it is possible to program data by sending the two-cycle continuous write command instead of the standard four-cycle write command.

Only submitting the “Reset from continuous mode” command changes the flash back to the normal mode. Other commands have no influence on the continuous mode setting, including Read/Reset and erase commands.

Asserting a flash hardware reset by asserting a CPU reset (Power reset, external reset, software reset, watchdog reset) or by writing to a protected sector also resets the flash from the continuous mode.

■ Example for writing to the flash memory

[Figure 54.8-1 Example of the flash memory write procedure \(Page No.1198\)](#) shows an example of the procedure for writing to the flash memory. The hardware sequence flags (see [section 54.7.3 Hardware](#)

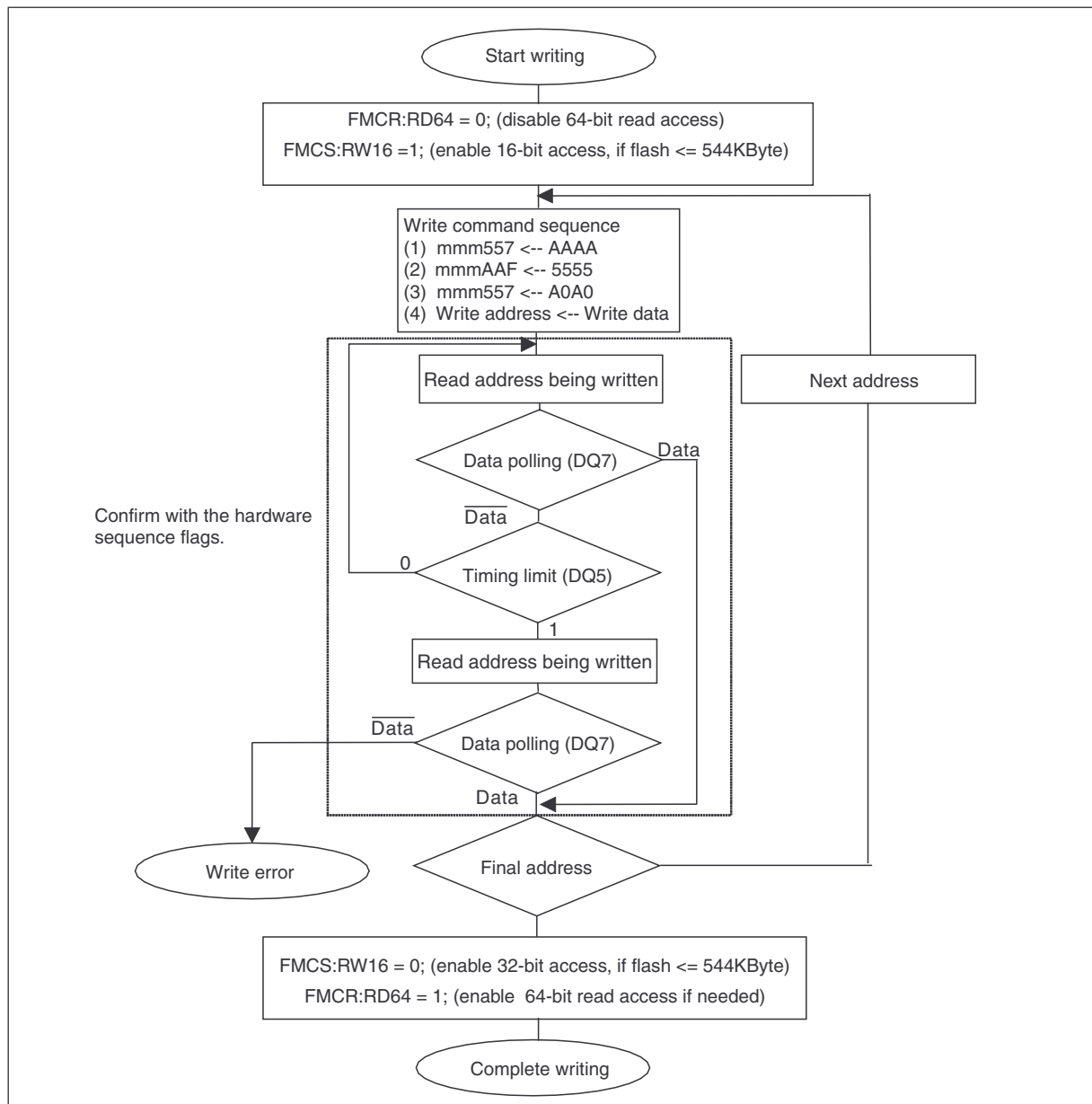
1. 32-bit flash access is not implemented on smaller devices. Please refer to [Table 54.6-1](#) or the device datasheets for possible flash access modes.

[Sequence Flags \(Page No.1189\)](#) can be used to determine the state of the automatic algorithm in the flash memory. Here, the data polling flag (DQ7) is used to confirm that writing has terminated.

Data polling must be performed to the memory location which is being programmed. Otherwise the transition of the DQ7 flag at the termination of the write command cannot be detected.

When the write automatic algorithm starts or stops, the Flash changes asynchronously between the automatic algorithm execution and the read/reset or sector erase suspend state. A possible read access to the Flash at this time can return invalid data. Hence the status of the hardware sequence flags must be rechecked to confirm the correct state.

Figure 54.8-1 Example of the flash memory write procedure



Note: The address "mmm" must point into the flash memory sector on which the command is meant to operate.

54.8.3 Erasing All Data (Chip Erase)

All data can be erased from a flash memory module by sending the Chip Erase command in the command sequence table (see [54.7.2 Auto Program Algorithm Commands \(Page No.1186\)](#)) continuously to any sector in the target flash memory module.

The Chip Erase command is executed in six bus operations. When writing of the sixth cycle is completed, the chip erase operation is started. For chip erase, the user does not need to write 0 to the flash memory before erasing. During execution of the automatic erase algorithm, the flash memory automatically writes 0 for verification before all of the cells are erased (preprogramming).

54.8.4 Erasing Optional Data (Sector erase)

Optional sectors in the flash memory can be erased by sending the Sector Erase command in the command sequence table (see [54.7.2 Auto Program Algorithm Commands \(Page No.1186\)](#)) continuously to the target sector in the flash memory. Individual sectors can be erased. Multiple sectors can also be specified at one time.

■ Specifying sectors

A Sector erase is initiated by submitting the sector erase command (six write operations) to the target sector. After submitting the last command (30h) to the target sector, the sector erase wait time is applied for approximately 50μs. To erase multiple sectors, write the erase code 30h (sixth cycle of the command sequence) to the next target sector within this wait time. The first 5 cycles of the sector erase command do not have to be written in this case.

■ Notes on specifying multiple sectors

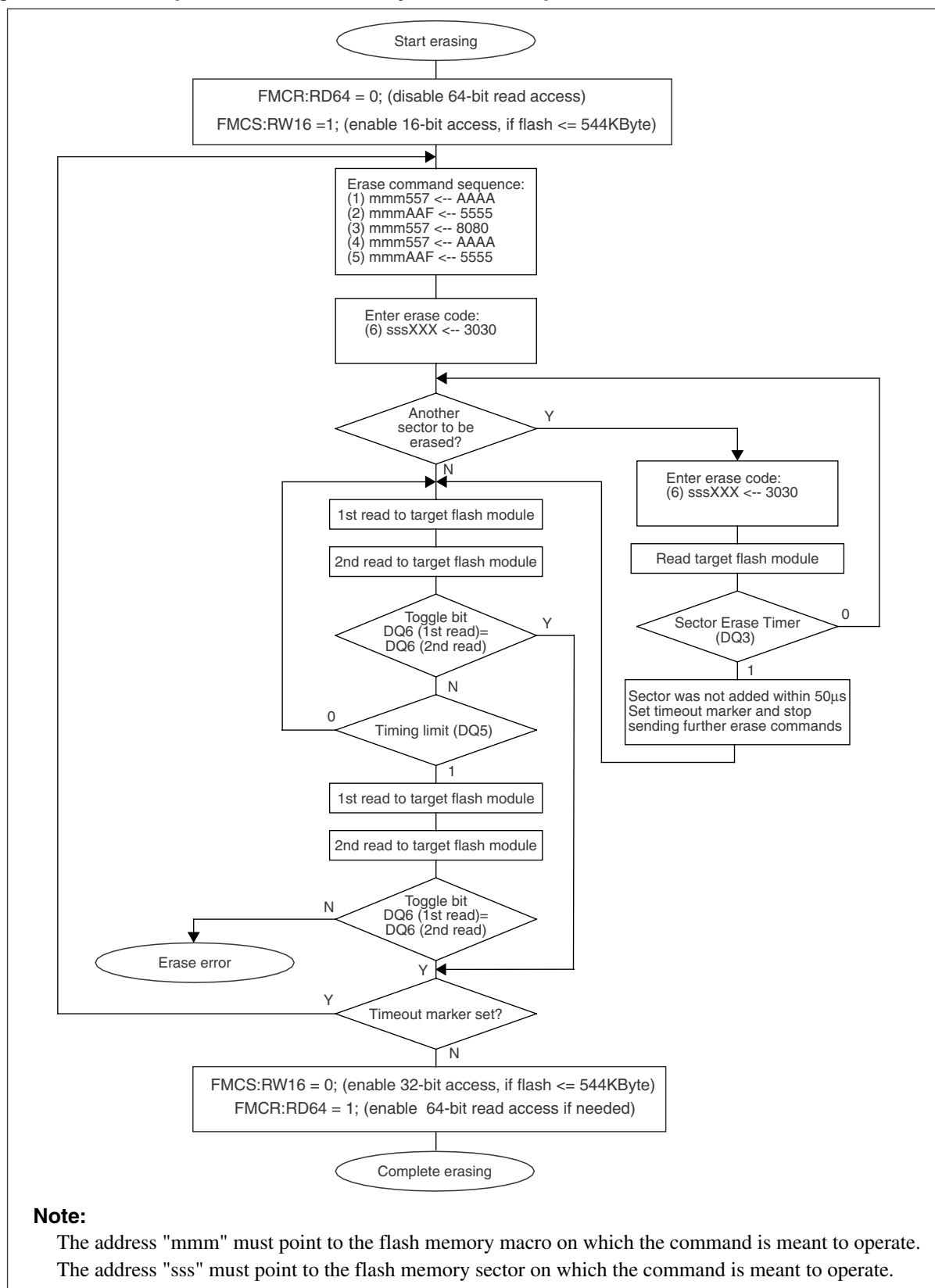
Erase is started when the sector erase wait period of 50μs terminates after the last sector erase code (30h) has been written. Each writing of a sector erase code restarts the sector erase wait period. The sector erase timer flag DQ3 must be checked after submitting each erase code to make sure it has been accepted.

■ Example for erasing sectors in the flash memory

The hardware sequence flags (see [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#)) can be used to determine the state of the automatic algorithm in the flash memory. [Figure 54.8-2 Example of the flash memory sector erase procedure \(Page No.1200\)](#) shows an example where the toggle bit flag (DQ6) is used to confirm that erasing has terminated.

When the automatic erase algorithm stops, the Flash changes asynchronously from the automatic algorithm execution and the read/reset state. A possible read access to the Flash at this time can return invalid data. For example to safely identify the Erase error state, the DQ6 toggle bits must be checked again after reading DQ5=1.

Figure 54.8-2 Example of the flash memory sector erase procedure



54.8.5 Suspending Sector Erase

Erasing of flash memory sectors can be suspended by sending the Sector Erase Suspend command (B0_H) to an address in the target flash memory module. Submitting this command suspends the sector erase operation being executed. In this suspend state, it is possible to read or write data from/to sectors that are not specified to be erased. Further erase operations are forbidden.

The Sector Erase Suspend command is valid only while the sector erase operation is in progress or while the sector erase wait time is being applied. The command will be ignored during chip erase or write operations. If a Sector Erase Suspend command is issued during sector erase suspend, the command will be ignored.

Entering the Sector Erase Suspend command during the sector erase wait period will immediately terminate sector erase wait and cancel the requested erase operation. While the sector erase is in progress, it takes a maximum period of 20ms from submitting the Sector Erase Suspend command until the flash changes to the suspend state.

■ Notes on using the sector erase suspend function of the Program/Data Flash (Main Flash):

- After starting or restarting a sector erase command, a minimum wait time of 1.2ms must be applied before submitting a sector erase suspend command.
- The erase operation of one sector must not be suspended for more than 10 times.

■ Notes on using the sector erase suspend function of the Data Flash:

The same rules as stated above for the Program/Data Flash (Main Flash) are also valid for the Data Flash. However it is allowed to suspend a sector erase operation more frequently, when the following rules are obeyed:

- During sector erase, the DQ4 flag must be polled with a maximum read period of 20μs.
- If DQ4 is 1 (busy to suspend), polling must continue.
- If DQ4 becomes 0 (ready to suspend), the erase suspend command must be written immediately.
- The maximum length of the "busy to suspend" period is 2.4ms. Within this time, no suspend command should be written to the flash.

54.8.6 Restarting Sector Erase

Suspended erasing of flash memory sectors can be restarted by sending the Sector Erase Restart command (30_H) to an address in the target flash memory module. If a Sector Erase Restart command is issued during sector erase, the command will be ignored.

The Sector Erase Restart command is used to restart erasing of sectors from the sector erase suspend state set by using the Sector Erase Suspend command.

54.9. Caution and Notes

■ Compatibility to MBM29LV200TC

For the Program/Data Flash (Main Flash), please review the Spansion MBM29LV200TC data sheet in conjunction with this document.

- The sector protect function of MBM29LV200TC via V_{ID} (12 V) pin is not supported. Instead, please refer to [Chapter 55 Flash Security \(Page No.1205\)](#).
- The Flash memory manufacturer code and device code are not available. The corresponding read commands specified for MBM29LV200TC are not supported.

■ Reset

A reset (Power on reset, external INITX, software reset, watchdog reset) asserts the flash hardware reset. Such a reset assertion during flash writing causes the data being written to be undefined.

Resetting the device once execution of sector erase has begun will corrupt the data in the sector. In that case, restart the erase on this sector and allow it to complete.

■ Program access to flash memory

When the automatic algorithm is operating, read access to the flash memory at which the automatic algorithm is active, is disabled.

Hence, make sure that the CPU is not executing code from a flash memory that is erased/written.

For the same reason, make sure that the table base register (TBR) is not pointing to an interrupt vector table located in flash memory to be erased or written to. Program TBR to point to an interrupt vector table in RAM, or disable interrupts completely before starting the automatic algorithm.

■ CPU mode

When in CPU mode, the address-allocation method is different than when writing via a parallel Flash programmer. Refer to [“54.3.2 Address conversion from CPU Mode to Flash Programming Mode \(Page No.1179\)”](#).

■ Flash memory mode (writing via parallel Flash programmer)

This Flash memory allows writing via an external device, by means of the parallel Flash programmer. In this state, pin functions equivalent to the stand-alone product Spansion MBM29LV400TC are assigned to the device's external pins, and operation of the CPU halts.

In Flash memory mode, the address line connections are changed from CPU mode, to mapping within the memory area.

See the section in the parallel Flash programmer manual concerning the MBM29LV400TC or MB91460 for details about using Flash memory from a parallel Flash programmer.

● Poweron Sequence in parallel programming mode

The flash memory can be accessed in programming mode after a certain wait time, which is needed for Security Vector fetch:

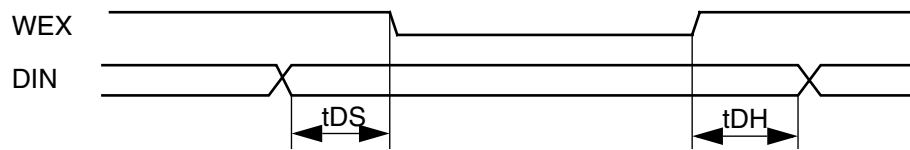
- Minimum wait time after VDD5/VDD5R power on: 2.76 ms
- Minimum wait time after INITX rising: 1.0 ms

● Timing Requirement of the Flash Security

To avoid unexpected cancellation of flash write sequences by the Flash Security, make sure that the write

MB91460 Series

data does not change during WEX=L. Check that the timing of write data DIN versus WEX is like shown here.

**■ Auto Program Algorithms and Flash Cache**

To maintain data consistency it is strongly recommended to disable the Flash Cache while writing to the flash memory and to flush the Flash Cache after completing the auto program algorithm.

MB91460 Series

Chapter 55 Flash Security

55.1. Overview

- Module for controlling the read and write access protection to the embedded Flash memory

55.2. Features

- Common read protection for all flash sectors depending on device mode
- Individual write protection for each flash sector depending on device mode
- Write protection level depending on device mode
- Security vector re-fetch sequence (for security status update after chip erase)
- CRC checksum calculation (CRC32/AAL5 algorithm)
- CRC polygon is $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

55.3. Flash Security Vectors

55.3.1 Vector addresses

Two Flash Security Vectors (FSV1, FSV2) are located parallel to the Boot Security Vectors (BSV1, BSV2) controlling the protection functions of the flash security module:

FSV1: 0x14:8000 BSV1: 0x14:8004

FSV2: 0x14:8008 BSV2: 0x14:800C

Remark: The addresses of both boot security vectors and flash security vectors depend on the size and the data width configuration of the embedded flash memory. Therefore always check the appropriate datasheet if the addresses shown here are valid for the corresponding product. The following tables explain the settings of the 1088 KByte flash macro, as used on MB91F467Dx.

55.3.2 Security Vector FSV1

The setting of the Flash Security Vector FSV1 is responsible for the read and write protection modes and the individual write protection of the 8 kByte sectors.

■ FSV1 (bits 31 to 16)

The setting of the Flash Security Vector FSV1 bits [31:16] is responsible for the read and write protection modes.

Table 55.3-1 Explanation of the bits in the Flash Security Vector FSV1[31:16]:

FSV1[31:19]	FSV1[18] Write Protec- tion Level	FSV1[17] Write Protec- tion	FSV1[16] Read Protec- tion	Flash Security Mode
set all to '0'	set to '0'	set to '0'	set to '1'	Read Protection (all device modes, except INTVEC mode MD[2:0]="000")
set all to '0'	set to '0'	set to '1'	set to '0'	Write Protection (all device modes, without exception)
set all to '0'	set to '0'	set to '1'	set to '1'	Read Protection (all device modes, except INTVEC mode MD[2:0]="000") and Write Protection (all device modes)
set all to '0'	set to '1'	set to '0'	set to '1'	Read Protection (all device modes, except INTVEC mode MD[2:0]="000")
set all to '0'	set to '1'	set to '1'	set to '0'	Write Protection (all device modes, except INTVEC mode MD[2:0]="000")
set all to '0'	set to '1'	set to '1'	set to '1'	Read Protection (all device modes, except INTVEC mode MD[2:0]="000") and Write Protection (all device modes except INTVEC mode MD[2:0]="000")

MB91460 Series

■ FSV1 (bits 15 to 0)

The setting of the Flash Security Vector FSV1 bits [15:0] is responsible for the individual write protection of the 8 kByte sectors. It is only evaluated if write protection bit FSV1[17] is set.

Table 55.3-2 Explanation of the bits in the Flash Security Vector FSV1[15:0] (valid for MB91F467Dx):

FSV1 bit	Sector	Enable Write Protection	Disable Write Protection	Comment
FSV1[0]	SA0	set to '0'	set to '1'	
FSV1[1]	SA1	set to '0'	set to '1'	
FSV1[2]	SA2	set to '0'	set to '1'	
FSV1[3]	SA3	set to '0'	set to '1'	
FSV1[4]	SA4	set to '0'	-	Write protection is mandatory !
FSV1[5]	SA5	set to '0'	set to '1'	
FSV1[6]	SA6	set to '0'	set to '1'	
FSV1[7]	SA7	set to '0'	set to '1'	
FSV1[8]	-	set to '0'	set to '1'	not available
FSV1[9]	-	set to '0'	set to '1'	not available
FSV1[10]	-	set to '0'	set to '1'	not available
FSV1[11]	-	set to '0'	set to '1'	not available
FSV1[12]	-	set to '0'	set to '1'	not available
FSV1[13]	-	set to '0'	set to '1'	not available
FSV1[14]	-	set to '0'	set to '1'	not available
FSV1[15]	-	set to '0'	set to '1'	not available

Remark: It is mandatory to always set the sector where the Flash Security Vectors FSV1 and FSV2 are located to write protected (here sector SA4). Otherwise it is possible to overwrite the Security Vector to a setting where it is possible to either read out the flash content or manipulate data by writing.

Remark: The content of the security vector FSV1[15:0] depends on the size and the data width configuration of the embedded flash memory. Therefore always check the appropriate datasheet if the data shown here are valid for the corresponding product.

See section [54.3.1 Flash Memory Sector Organisation \(Page No.1177\)](#) for an overview about the sector organization of the Flash Memory.

55.3.3 Security Vector FSV2

The setting of the Flash Security Vector FSV2 bits [31:0] is responsible for the individual write protection of the 64 kByte sectors. It is only evaluated if write protection bit FSV1[17] is set.

Table 55.3-3 Explanation of the bits in the Flash Security Vector FSV2[31:0] (valid for MB91F467Dx):

FSV2 bit	Sector	Enable Write Protection	Disable Write Protection	Comment
FSV2[0]	SA8	set to '0'	set to '1'	
FSV2[1]	SA9	set to '0'	set to '1'	
FSV2[2]	SA10	set to '0'	set to '1'	
FSV2[3]	SA11	set to '0'	set to '1'	
FSV2[4]	SA12	set to '0'	set to '1'	
FSV2[5]	SA13	set to '0'	set to '1'	
FSV2[6]	SA14	set to '0'	set to '1'	
FSV2[7]	SA15	set to '0'	set to '1'	
FSV2[8]	SA16	set to '0'	set to '1'	
FSV2[9]	SA17	set to '0'	set to '1'	
FSV2[10]	SA18	set to '0'	set to '1'	
FSV2[11]	SA19	set to '0'	set to '1'	
FSV2[12]	SA20	set to '0'	set to '1'	
FSV2[13]	SA21	set to '0'	set to '1'	
FSV2[14]	SA22	set to '0'	set to '1'	
FSV2[15]	SA23	set to '0'	set to '1'	
FSV2[16]	-	set to '0'	set to '1'	not available
FSV2[17]	-	set to '0'	set to '1'	not available
FSV2[18]	-	set to '0'	set to '1'	not available
FSV2[19]	-	set to '0'	set to '1'	not available
FSV2[20]	-	set to '0'	set to '1'	not available
FSV2[21]	-	set to '0'	set to '1'	not available
FSV2[22]	-	set to '0'	set to '1'	not available
FSV2[23]	-	set to '0'	set to '1'	not available
FSV2[24]	-	set to '0'	set to '1'	not available
FSV2[25]	-	set to '0'	set to '1'	not available
FSV2[26]	-	set to '0'	set to '1'	not available
FSV2[27]	-	set to '0'	set to '1'	not available
FSV2[28]	-	set to '0'	set to '1'	not available
FSV2[29]	-	set to '0'	set to '1'	not available
FSV2[30]	-	set to '0'	set to '1'	not available
FSV2[31]	-	set to '0'	set to '1'	not available

Remark: The content of the security vector FSV2[31:0] depends on the size and the data width configuration of the embedded flash memory. Therefore always check the appropriate datasheet if the data shown here are valid for the corresponding product.

See section [54.3.1 Flash Memory Sector Organisation \(Page No.1177\)](#) for an overview about the sector organization of the Flash Memory.

MB91460 Series

55.4. Register

55.4.1 Flash Security Control Register

■ FSCR0: Address 7100h (Access: Byte (write), Word (read))

31	30	29	28	27	26	25	24	bit
CRC31/ S7	CRC30/ S6	CRC29/ S5	CRC28/ S4	CRC27/ S3	CRC26/ S2	CRC25/ S1	CRC24/ S0	
1	1	1	1	1	1	1	1	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
23	22	21	20	19	18	17	16	bit
CRC23	CRC22	CRC21	CRC20	CRC19	CRC18	CRC17	CRC16	
1	1	1	1	1	1	1	1	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R	R	R	R	R	R	R	R	Attribute
15	14	13	12	11	10	9	8	bit
CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8	
1	1	1	1	1	1	1	1	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R	R	R	R	R	R	R	R	Attribute
7	6	5	4	3	2	1	0	bit
CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0	
1	1	1	1	1	1	1	1	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R	R	R	R	R	R	R	R	Attribute

(See “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)” for details of the attributes.)

• Bit31-24: Sequence activation

S7-S0	Function
0xA5 -> 0x5A	Start of a Flash Security Vector Re-Fetch Sequence (write only)
0xF0 -> 0x0F	Start of a Flash Memory CRC32 Checksum Sequence (write only)

Continuously writing “A5_H”, “5A_H” in the FSCR0[31:24] register will start a Flash Security Vector Re-fetch sequence immediately after writing “5A_H”. There is no time restrictions between “A5_H” and “5A_H”, but if “A5_H” is written followed by the one other than “5A_H”, it must be written “A5_H” again. If not, the Re-Fetch sequence cannot be started even if “5A_H” is written.

Continuously writing “F0_H”, “0F_H” in the FSCR0[31:24] register will start a CRC32 checksum sequence immediately after writing “0F_H”. There is no time restrictions between “F0_H” and “0F_H”, but if “F0_H” is written

followed by the one other than “0F_H”, it must be written “F0_H” again. If not, the CRC checksum sequence cannot be started even if “0F_H” is written.

Remark: The Flash Security Vector Re-Fetch sequence is especially intended to be used after a chip erase command to update the security status without the need of applying an external INITX reset or after changing the status of the FSV1 security vector.

Remark: For both Security Vector Re-Fetch and CRC32 it is not recommended to start these sequences from programs located in the Flash Memory itself.

Remark: Before starting the CRC32 checksum it is recommended to set the RC oscillation in the 2MHz operation mode to avoid long runtimes. See the RCSEL bit in [Chapter 48 Clock Monitor Configuration Register \(Page No.1125\)](#).

- **Bit31-0:** CRC32 checksum result

CRC31-CRC0	Function
	CRC32 Checksum (read only)

This register contains the CRC32 checksum result after completion of the CRC32 checksum sequence (the sequence completion is indicated by FSCR1.RDY). The CRC checksum is calculated in a standard CRC32/AAL5 algorithm with the polygon $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$.

MB91460 Series

■ FSCR1: Address 7104h (Access: Byte (read), Word (write))

31	30	29	28	27	26	25	24	bit
SVF_RDY	-	-	-	-	-	-	RDY	
0	0	0	0	0	0	0	0	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R	RX/WX	RX/WX	RX/WX	R0/WX	R0/WX	R0/WX	R	Attribute
23	22	21	20	19	18	17	16	bit
-	-	-	-	CSZ3	CSZ2	CSZ1	CSZ0	
0	0	0	0	0	0	0	0	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W	Attribute
15	14	13	12	11	10	9	8	bit
CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
0	0	0	0	0	0	0	0	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute
7	6	5	4	3	2	1	0	bit
CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
0	0	0	0	0	0	0	0	Initial value (INIT pin input, watchdog reset)
X	X	X	X	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

(See “Meaning of Bit Attribute Symbols (Page No.15)” for details of the attributes.)

- **Bit31:** SVF_RDY: Security Vector Fetch Ready

SVF_RDY	Function
0	Security vector fetch sequence running or not yet started
1	Security vector fetch sequence is ready

This bit is cleared by:

- external reset (INITX pin = 0) or watchdog reset,
- writing the sequence 0xA5 -> 0x5A to FSCR0[31:24].

Both events will start a security vector refetch sequence.

SVF_RDY is set when the refetch sequence is finished and the newly read security data is valid.

- **Bit30-25:** Reserved bit. The read value is always “0”.

- **Bit24:** RDY: CRC32 Sequence Ready

RDY	Function
0	CRC32 sequence running or not yet started
1	CRC32 sequence ready (data in the FSCR0 register is valid)

- **Bit23-20:** Reserved bit. The read value is always "0".
- **Bit19-16:** CSZ3-0: CRC32 Size Mask

CSZ3-0	Function
0000	CRC32 sequence size mask is 4 kByte
0001	CRC32 sequence size mask is 8 kByte
0010	CRC32 sequence size mask is 16 kByte
0011	CRC32 sequence size mask is 32 kByte
0100	CRC32 sequence size mask is 64 kByte
0101	CRC32 sequence size mask is 128 kByte
0110	CRC32 sequence size mask is 256 kByte
0111	CRC32 sequence size mask is 512 kByte
1000	CRC32 sequence size mask is 1024 kByte
1001	CRC32 sequence size mask is 2048 kByte
1010	CRC32 sequence size mask is 4096 kByte
1011-1111	Not supported

Remark: CSZ3-0 is used as an OR-mask for the address given by CSA15-0. See address calculation below.

- **Bit15-0:** CSA15-0: CRC32 Start Address

This register contains the CRC32 start address which is aligned to 4kByte addresses. It is only possible to calculate the CRC32 checksum over addresses located in the Flash Memory address space. Other addresses are invalid and might lead to wrong checksums.

Remark: The addresses to be written in this register are flash memory addresses like used in the flash parallel programming mode and not the mapped addresses which are used in CPU mode.

■ Calculation of the CRC32 start- and end-address:

The CSZ3-0 setting is first translated into a mask value:

CSZ3-0	MASK
0000	0000_0000_0000_0000
0001	0000_0000_0000_0001
0010	0000_0000_0000_0011
0011	0000_0000_0000_0111
0100	0000_0000_0000_1111
0101	0000_0000_0001_1111
0110	0000_0000_0011_1111
0111	0000_0000_0111_1111
1000	0000_0000_1111_1111
1001-1111	and so on...

CRC32 Start address = CSA[15:0] << 12 + 0x000

CRC32 End address = (CSA[15:0] or MASK) << 12 + 0xFFFF

MB91460 Series

Chapter 56 Embedded Data Flash

MB91FV460B and MB91F467P contain a 64 KByte internal data flash.

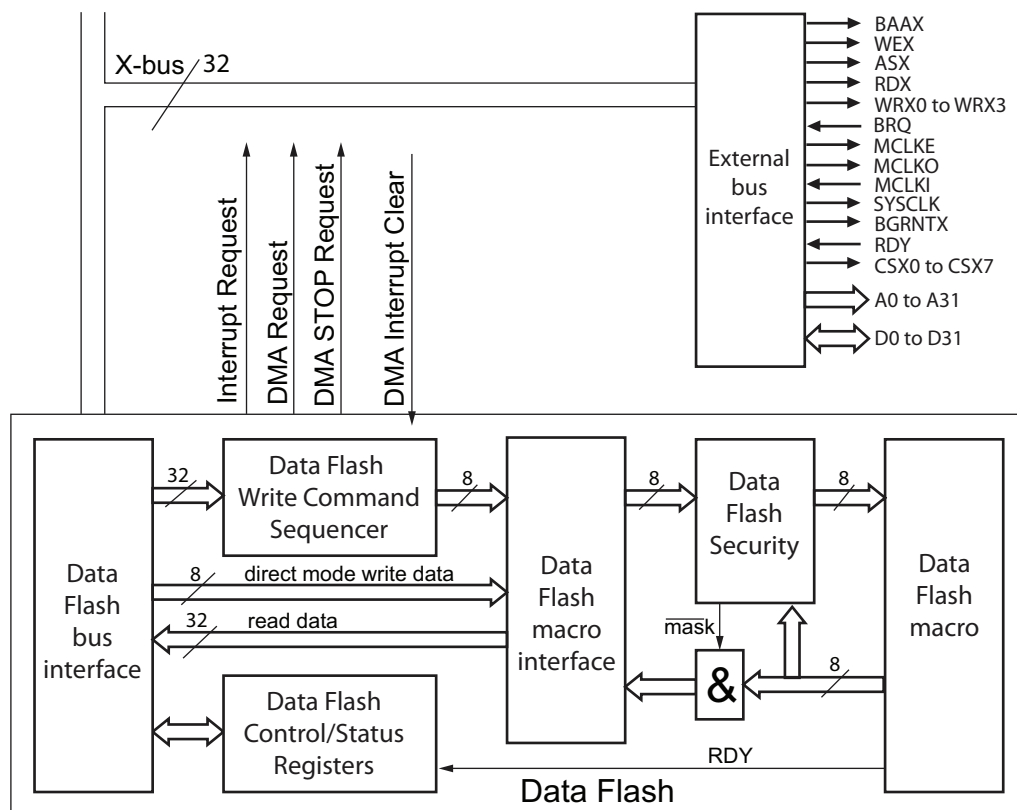
56.1. Data Flash Features

- Size of flash macro: 64 Kbytes (4 x16 Kbytes + 1 x 256 bytes security sector)
- Data width of flash macro: 8 bit
- Synchronous flash interface and flash macro
- 2 access modes (direct access, command sequencer access)
- Read access 8/16/32-bit by internal sequencer hardware
- Write access 8-bit in direct access mode, 8/16/32-bit in command sequencer write mode
- Programmable wait states for read/write access
- Data Flash Security feature (read and write protection)
- CRC calculation feature
- Interrupt- and DMA request, DMA stop request

56.2. Data Flash Block Diagram

The Data Flash consists of the flash macro and interface, control, status, command sequencer and security logic.

On MB91460 series devices, the Data Flash is connected to the X-Bus in parallel to the External Bus interface:



56.3. Data Flash Operation Modes

The data flash is located in the top address space of external bus area. Per **default** (after software reset RST), the data flash is **disabled** and does not accept any read/write access. The data flash can be enabled by setting the bit DFCS:FLASHEN (DFCS is the Data Flash Control/Status register).

56.3.1 Direct Access mode:

The Direct Access mode provides data flash access similar to the access of the embedded program/data flash (main flash). For write/program operations, the flash command sequences must be written by the CPU. The command sequences are the same as used for the embedded program/data flash (main flash).

- CPU reads data in byte, halfword or word (8/16/32-bit) length units, whereas 16- or 32-bit read operations are split into 2 or 4 sequential 8-bit flash macro read accesses by hardware.
- CPU writes data in byte (8-bit) width units.
- For write/program operations, the flash command sequences must be written by the CPU.
- The flash macro auto algorithms (Chip Erase, Sector Erase, Sector Erase Suspend,...) can only be activated in direct access mode.
- Direct access mode is the default mode after software reset (RST).

56.3.2 Command Sequencer Mode:

In command sequencer mode, the flash macro command sequences for data write operation are generated by hardware.

- CPU reads data in byte, halfword or word (8/16/32-bit) length units (same as in direct access mode).
- CPU writes data in byte, halfword or word (8/16/32-bit) length units using normal “store” instructions. The flash macro command sequences are generated by internal command sequencer hardware. For 16- or 32-bit write, 2 or 4 command sequences are generated, respectively.
- The data flash interface will not issue wait states after a command sequencer write operation was started. The CPU can continue working during data flash programming.
- **If a command sequencer write operation is ongoing, and the CPU writes data again, this second write request is ignored!** The error flag DFWS:PAERF is set in case of such a prohibited access. It is recommended to use the data flash interrupts, which indicate that the preceeding write sequence was finished and successful.
- If a command sequencer write operation is ongoing, and the CPU tries to read data, 0x00 is returned and the error flag DFWS:PAERF is set.
- The flash macro auto algorithms (Chip Erase, Sector Erase, Sector Erase Suspend,...) cannot be activated.
- Command Sequencer mode is enabled by setting the bit DFWC:WE (Data Flash Write Control register).
- After software reset (RST), the command sequencer mode is disabled.

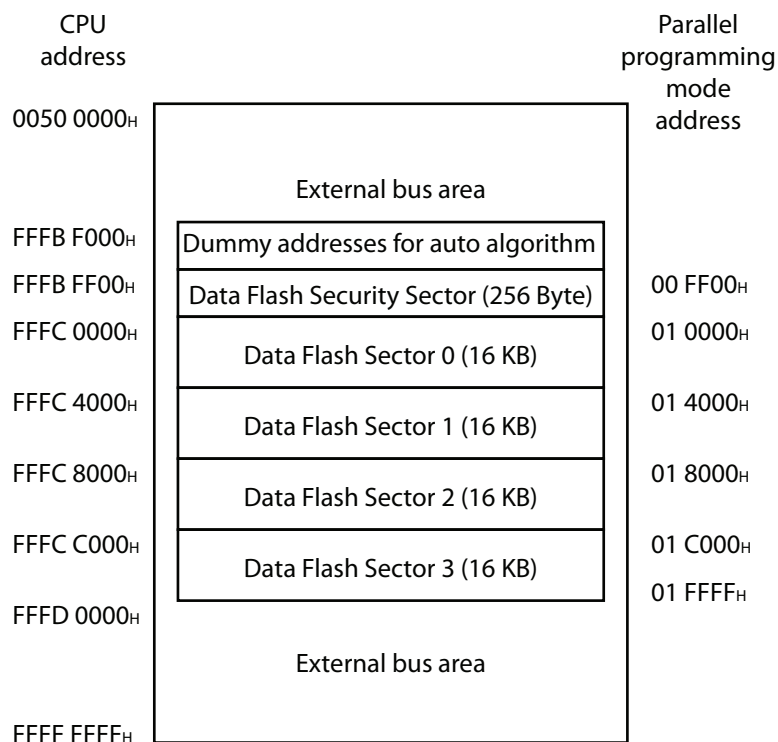
56.3.3 Parallel Programming mode:

- The parallel programming mode works similar to the main flash memory. The function/timing of some external control lines are different.
- In parallel programming mode, it is not necessary to set the Data Flash enable bit (DFCS:FLASHEN).
- Data Flash Memory access is performed in byte (8-bit) length units.

56.4. Data Flash access in CPU mode

56.4.1 Data Flash Memory Map

The Data Flash macro is 8 bit wide. It is located in the top address space of external bus area:



Note: The address in parallel programming mode is listed here without 10:0000h offset.

Set the offset by keeping FA[22:20] = 001 the same kind as used for programming of the main flash.

Note: The “Dummy addresses for auto algorithm” are accepted although they are located below the physical addresses of the flash macro. This address space is needed to apply correct addresses in auto algorithms. See the example in [“Auto Program Algorithms” on page 1216](#) . However, toggle flags cannot be read using the dummy addresses.

56.4.2 Data Flash and External Bus

If the Data Flash is disabled (see [“Data Flash Operation Modes” on page 1214](#)), the complete address space can be used for the external bus.

If the Data Flash is enabled, the user should take care that no external bus chip select area overlaps the address range of the Data Flash.

If a chip select area overlaps the Data Flash addresses, the following scenario may appear:

- Write operations will be sent to data flash and external bus in parallel. This may cause heavy problems, especially if the data flash is written in direct mode, where the CPU sends the command sequences for programming (see [“Direct Access mode:” on page 1214](#)).
- Read operations will return unpredictable results.

56.4.3 Flash access timing settings in CPU mode

The Data Flash can be accessed up to CLKB = 100 MHz. For timing and wait state setup, please refer to the description of the bits TMG2, TMG1, TMG0 in [“Data Flash Control and Status Register” on page 1219](#) .

Although the data flash is located in the address space of external bus, there is no dependency between external bus timing and data flash timing.

56.4.4 Auto Program Algorithms

The auto program algorithms can only be applied in direct access mode, while the “Program” sequence can be generated by hardware if the Command Sequencer Mode is used.

The data flash supports command sequences similar to the main flash:

Table 56.4-1 Data flash command sequences

Command Sequence	Bus Write Cycle	1st bus Write cycle		2nd bus Write cycle		3rd bus Write cycle		4th bus Write cycle		5th bus Write cycle		6th bus Write cycle	
		Address	Data	Address	Data	Address	Data			Address	Data	Address	Data
Read/Reset	1	XXX	F0	RA	RD								
Read/Reset	3	AA8	AA	554	55	AA8	F0	RA	RD				
Program	4	AA8	AA	554	55	AA8	A0	PA	PD				
Chip Erase	6	AA8	AA	554	55	AA8	80	AA8	AA	554	55	AA8	10
Sector Erase	6	AA8	AA	554	55	AA8	80	AA8	AA	554	55	SA	30
Sector Erase Suspend		Sector Erase Suspend by input of address “XXX” and data “B0”											
Sector Erase Resume		Sector Erase Resume by input of address “XXX” and data “30”											
Unlock Bypass set	3	AA8	AA	554	55	AA8	20						
Unlock Bypass program	2	XXX	A0	PA	PD								
Unlock Bypass Reset	2	XXX	90	XXX	F0/00								

PA: Program Address

PD: Program Data. Data to be programmed at location PA.

RA: Read Address

RD: Data to read at location RA.

SA: Sector Address (points into the sector to be erased)

It is recommended that the addresses “AA8” and “554” point into the sector which is to be programmed.

For **example**, to program a byte into sector SAS, the following sequence should be used:

Address PA=0xFFFFBFF83 is inside sector SAS.

1. write addr=0xFFFFBFAA8 data=0xAA
2. write addr=0xFFFFBF554 data=0x55
3. write addr=0xFFFFBFAA8 data=0xA0
4. write addr=0xFFFFBFF83 =PA data=PD

Note: The address for the write sequence (1., 2., 3. write) points into the “Dummy addresses for auto algorithm” here. For polling of toggle bits, an address pointing inside the programmed sector has to be used, for example the programmed address (PA) itself.

MB91460 Series**56.4.5 Data Flash Hardware Sequence Flags (Toggle Bits)**

In direct access mode, the data flash returns toggle bits shown in the following table.

In command sequencer mode, it is not necessary to read the toggle bits because they are observed by the command sequencer automatically.

Table 56.4-2 Data flash hardware sequence flags

Status		Accessed address	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2
Normal operation	Read/Reset	Any address	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	Write (embedded program algorithm running)	Address being programmed	DATA:7	Toggle	0	0	0	1
	Sector erase wait (Flash waits for inputting further sector erase requests)	Sectors which are already specified for erase	1	Toggle	0	0	0	Toggle
		Other sectors						1
	Chip/sector erase (embedded erase algorithm running)	Sectors which are specified for erase (all sectors in case of Chip erase)	0	Toggle	0	1: busy to suspend 0: ready to suspend	1	Toggle
		Other sectors						1
	Sector erase suspended	Sectors which are specified for erase	1	1	0	0	0	Toggle
		Other sectors	DATA:7	DATA:6	DATA:5	DATA:4	DATA:3	DATA:2
	Writing of another sector while Flash is in erase suspended state	Address being programmed	DATA:7	Toggle	0	0	0	1
	Chip/sector erase finished (Read/Reset state)	Sector which was erased	DATA:7 = 1	DATA:6 = 1	DATA:5 = 1	DATA:4 = 1	DATA:3 = 1	DATA:2 = 1
Timing limit exceeded	Write	Address being programmed	DATA:7	Toggle	1	0	0	1
	Chip/sector erase	Sector which caused the Timeout during erase	0	Toggle	1	undefined	1	Toggle
		Other sectors						1

Note: For polling of toggle bits, an address pointing inside the programmed sector has to be used, for example the programmed address itself. Do not use a “Dummy addresses for auto algorithm”.

For a detailed description of the hardware sequence flags, please refer to section [54.7.3 Hardware Sequence Flags \(Page No.1189\)](#) in [Chapter 54 Flash Memory](#).

56.5. Data Flash Registers

The Data Flash has the following control/status registers:

DFCS : Data Flash Control and Status Register

Address	31	30	29	28	27	26	25	24	bit
07114 _H	RDYI	TMG2	TMG1	TMG0	FLASHEN	INTE	RDYINT	RDY	
	0	1	1	1	0	0	0	1	Initial value
	R/W	R/W	R/W	R/W	R/W	R/W	R/W0	R	Attribute

DFWC : Data Flash Write Command Sequencer Control Register

Address	23	22	21	20	19	18	17	16	bit
07115 _H	-	-	-	ERINTE	FININTE	IDLINTE	IDLMAE	WE	
	x	x	x	0	0	0	0	0	Initial value
	-	-	-	R/W	R/W	R/W	R/W	R/W	Attribute

DFWS : Data Flash Write Command Sequencer Status Register

Address	15	14	13	12	11	10	9	8	bit
07116 _H	PAERF	WIERINT	WERINT	TOERINT	FININT	IDLINT	ST1	ST0	
	0	0	0	0	0	0	0	0	Initial value
	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R	R	Attribute

DFSCR0 : Data Flash Security Control Register 0

Address	31:24	23:16	15:8	7:0	bit
07118 _H	DFSCR0[31:0]				
	1111 1111	1111 1111	1111 1111	1111 1111	Initial value
	R/W, R	R	R	R	Attribute

DFSCR1 : Data Flash Security Control Register 1

Address	31:24	23:16	15:8	7:0	bit
0711C _H	DFSCR1[31:0]				
	0 - - - 0001	0000 0000	0000 0000	0000 0000	Initial value
	R, R/W	R, R/W	R, R/W	R, R/W	Attribute

MB91460 Series**56.5.1 Data Flash Control and Status Register**

This section explains the Data Flash Control and Status register.

Addr: 0x07114

DFCS : Data Flash Control and Status register

31	30	29	28	27	26	25	24	bit
RDYI	TMG2	TMG1	TMG0	FLASHEN	INTE	RDYINT	RDY	
0	1	1	1	0	0	0	1	initial
R/W	R/W	R/W	R/W	R/W	R/W	R/W0	R	attribute

RDYI	Ready Inversion
0	(default) Normal flash operation
1	Setting this bit to '1' activates the RDYI input of the Flash. As a result, the RDY output of the Flash goes to '0' (used for test purposes only). Always write 0 to this bit.

- This bit is cleared by software reset (RST).
- Always write 0 to this bit.

TMG2	TMG1	TMG0	CLKB Frequency up to	Nr. of CLKB cycles per...		
				Read Operation	Write Operation (direct mode)	Write Operation (Command Sequencer mode)
0	0	0	6.2 MHz	3	3	6
0	0	1	16.7 MHz	4	3	6
0	1	0	33.3 MHz	5	3	6
0	1	1	50 MHz	6	3	6
1	0	0	66.6 MHz	8	4	6
1	0	1	83.3 MHz	9	5	6
1	1	0	100 MHz	10	6	6
1	1	1	(default) 100 MHz	11	6	6

- These bits control the number of wait cycles for read and write operations.
- The bits are set to "111" by software reset (RST) and can be read and written
- The Command Sequencer always uses 6 CLKB cycles per write operation

FLASHEN	Data Flash Enable
0	(default) Data Flash is disabled and does not accept read and write access
1	Data Flash is enabled and can be read and written depending on data flash security settings.

- This bit is cleared by software reset (RST) and can be read and written
- Before setting this bit, the user has to take care that no External Bus Chip Select area overlaps the data flash address space.

INTE	Ready Interrupt Enable
0	(default) Disable the interrupt of the RDYINT flag
1	Enable the interrupt of the RDYINT flag

- If this bit is cleared, no interrupt is generated when the RDYINT flag is set.
- If this bit is set, the interrupt by RDYINT flag is enabled.
- This bit is cleared by software reset (RST) and can be read and written.

RDYINT	Ready Interrupt Flag
0	(default) The flash macro has not entered the READY state
1	The flash macro has entered the READY state

- This bit is set after a rising edge of the RDY status line of the flash macro.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- Read-modify-write operations will read 1.

RDY	Flash Macro Ready Status
0	Indicates that a program/erase command is currently executed. Only the reset and suspend commands are accepted in this state.
1	(default) Indicates that no program/erase command is currently executed. Any command can be written to the Flash.

- This bit shows the RDY status line of the flash macro, sampled with 2 cycles of CLK_B. The RDY line of the flash macro switches to low level after a certain response time t_{BUSY} and switches back to high level after the programming/chip erase/sector erase time.
In direct access mode, t_{BUSY} is minimum 90 ns after the last write access of a program sequence.
In write sequencer mode, the command sequencer cares about RDY signal. There is no need to poll RDY.
- This bit is read-only.

56.5.2 Data Flash Write Command Sequencer Control Register

This section explains the Data Flash Sequencer Control register

Addr: 0x07115 DFWC : Data Flash Write Command Sequencer Control

23	22	21	20	19	18	17	16	bit
-	-	-	ERINTE	FININTE	IDLINTE	IDLMAE	WE	
x	x	x	0	0	0	0	0	initial
-	-	-	R/W	R/W	R/W	R/W	R/W	attribute

- Always write 0 to the bits 7:5.

ERINTE	Error Interrupt Enable
0	(default) Disable the interrupt of the error flags
1	Enable the interrupt of the error flag

- If this bit is cleared, no interrupt is generated when a error flag (TOERINT, WERINT and WIERINT) is set.
- If this bit is set, an interrupt is generated when one of the error flags is set.

MB91460 Series

- This bit is cleared by software reset (RST) and can be read and written.

FININTE	Finish Interrupt Enable
0	(default) Disable the interrupt of the FININT flag
1	Enable the interrupt of the FININT flag

- If this bit is cleared, no interrupt is generated when the FININT flag is set.
- If this bit is set, an interrupt is generated when the FININT flag is set.
- This bit is cleared by software reset (RST) and can be read and written.

IDLINTE	Idle Interrupt Enable
0	(default) Disable the interrupt of the IDLINT flag
1	Enable the interrupt of the IDLINT flag

- If this bit is cleared, no interrupt is generated when the IDLINT flag is set.
- If this bit is set, an interrupt is generated when the IDLINT flag is set.
- This bit is cleared by software reset (RST) and can be read and written.

IDLMAE	Idle DMA Enable
0	(default) Disable the DMA transfer request
1	Enable the DMA transfer request if the IDLINT flag is set

- If this bit is cleared, no DMA transfer request is generated when the IDLINT flag is set.
- If this bit is set, an DMA transfer request is generated when the IDLINT flag is set.
- This bit is cleared by software reset (RST) and can be read and written.

WE	Write Command Sequencer Enable
0	(default) Disable the Write Command Sequencer, Data Flash operates in direct mode
1	Enable the Write Command Sequencer Mode

- This bit enables the Command Sequencer mode.
- This bit is cleared by software reset (RST) and can be read and written.
- Do not clear this bit during a write sequence is ongoing! The write sequence would be killed immediately.

56.5.3 Data Flash Write Command Sequencer Status Register

This section explains the Data Flash Command Sequencer Status register.

Addr: 0x07116 DFWS : Data Flash Write Command Sequencer Status

15	14	13	12	11	10	9	8	bit
PAERF	WIERINT	WERINT	TOERINT	FININT	IDLINT	ST1	ST0	
0	0	0	0	0	0	0	0	initial
R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R	R	attribute

The command sequencer status flags are only set if the command sequencer is enabled (DFWC:WE=1).

PAERF	Prohibited Access Error Flag
0	(default) No prohibited access detected
1	Prohibited access detected

- This flag is set if the CPU tried to read or write into the Data Flash area while the Data Flash is accessed by the Command Sequencer.
- This flag cannot generate an interrupt.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- Read-modify-write operations will read 1.

WIERINT	Write Incomplete Error Flag
0	(default) Command sequencer write operation was completed
1	Command sequencer was disabled while a write operation was ongoing

- This flag is set when the command sequencer is disabled (set DFWC:WE=0) in "not idle" state .
- If this flag is 0, it is no guarantee that the write operation was successful. Use the FININT flag!
- This flag can generate an interrupt if DFWC:ERINTE is set.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- Read-modify-write operations will read 1.

WERINT	Write Error Flag
0	(default) No write error detected
1	Write operation returned error

- This flag is set after a write access returned error:
 - tried to write to an erase-suspended or write-protected sector,
 - tried to write a bit "1" although it is already "0" in flash.
- This flag can generate an interrupt if DFWC:ERINTE is set.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- Read-modify-write operations will read 1.

TOERINT	Timeout Error Flag
0	(default) No timeout error detected
1	A write operation ended with timeout error

- This flag is set after a write operation ended in timeout state.
- This flag can generate an interrupt if DFWC:ERINTE is set.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- Read-modify-write operations will read 1.

FININT	Command Sequence Finished Flag
0	(default) Write command was not (yet) finished successfully
1	Write command was finished successfully

- This flag is set after a command sequencer write operation was finished successfully.
- This flag can generate an interrupt if DFWC:FININTE is set.

- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- This bit is also cleared after a DMA transfer (caused by IDLINT) was finished.
- Read-modify-write operations will read 1.

IDLINT	Command Sequencer Idle Flag
0	(default) Command sequencer is disabled or not in IDLE state
1	Command sequencer entered the IDLE state

- This flag is set after the command sequencer was enabled (set DFWC:WE=1) or entered the IDLE state after a write operation was finished.
- This flag can generate an interrupt if DFWC:INTE is set.
- This flag can generate a DMA transfer request if DFWC:IDLDMAE is set.
- This bit is cleared by software reset (RST) or by writing 0. Writing 1 has no effect.
- This bit is also cleared after a DMA transfer was finished.
- Read-modify-write operations will read 1.

ST1	ST0	Command Sequencer Status Flags
0	0	(default) Command sequencer is disabled or in IDLE state
0	1	Command sequencer is submitting the write command
1	0	Command sequencer is waiting for Flash program finish
1	1	Command sequencer was disabled in "not idle" state

- Status bit {ST1,ST0} =2'b11 show that the command sequencer was disabled in "not idle" state and direct access to Flash is not yet permitted (wait for preceeding Flash sequence to finish). Max duration of this wait can be 11 clock cycle after disabling Command Sequencer.

56.5.4 Data Flash security Control Register 0,1

Please refer to [“Data Flash Security Registers” on page 1228](#) .

56.6. Data Flash Interrupts and DMA Access

If a command sequencer write operation is ongoing, and the CPU writes data again, this second write request is ignored! Therefore, it is recommended to use the data flash interrupts or DMA, which indicates that the write sequence is finished and successful.

56.6.1 Data Flash Interrupt Flag Overview

The Data Flash interface has 6 interrupt flags with certain relationship to the 3 output lines for interrupt / DMA request:

Interrupt Flags:

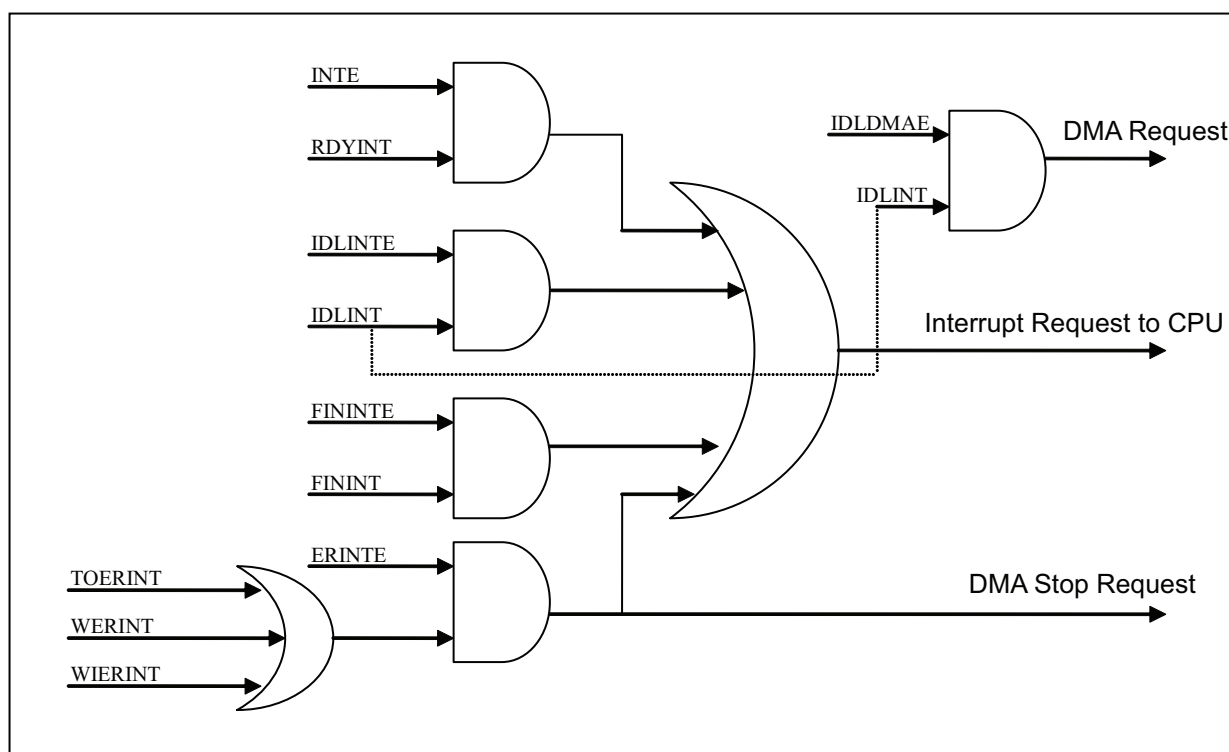
- IDLINT IDLE flag, indicates that the command sequencer has entered the IDLE state after a write sequence. This flag is also set just after the command sequencer was enabled by setting DFWC:WE.
- RDYINT READY flag, indicates that the flash macro has entered READY state.
- FININT FINISH flag, indicates that the command sequencer finished a write sequence successfully.
- TOERINT TIMEOUT Error flag, indicates that a command sequencer write sequence ended in TIMEOUT error state.
- WERINT Suspend Sector Write Error flag, indicates that there was a write request to a sector which is

erase suspended or write-protected and not ready for writing.

- **WIERINT** Write Incomplete Error flag, indicates that the command sequencer was disabled (DFWC:WE = 0) while a write sequence was ongoing.
- **PAERF** Prohibited Access Error flag, indicates that the CPU tried a read or write access while a command sequencer write was ongoing. PAERF is a status flag and cannot generate an interrupt.

The following picture shows the interrupt flags and their enable bits.

Figure 56.6-1 Data flash interrupts



The DMA request can be activated by the IDLE flag only and has a separate enable bit (DFWC:IDLDMAE). DMA Stop request is activated by the error flags.

The CPU interrupt can be activated by all interrupt flags.

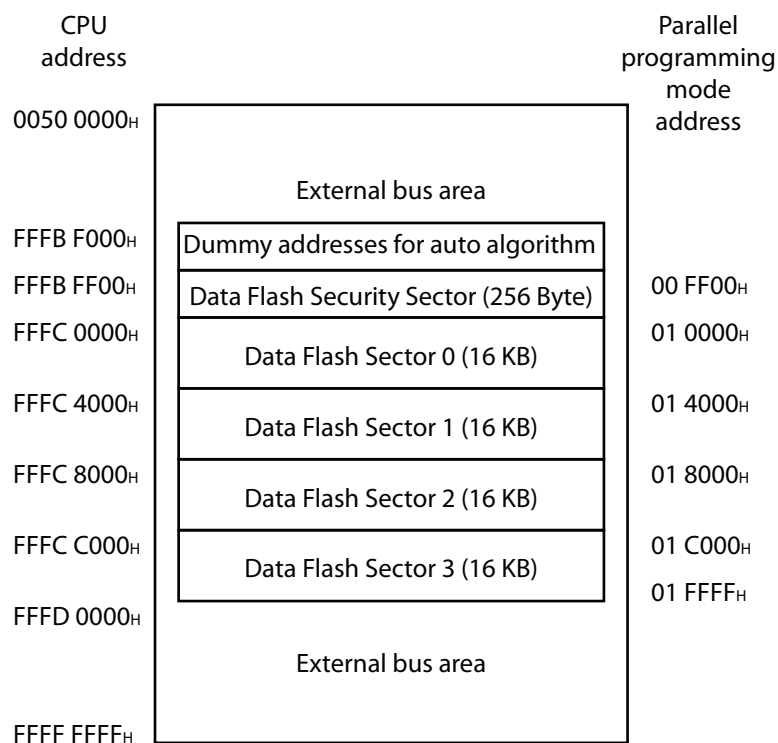
56.7. Data Flash parallel programming mode

Note: The currently available parallel flash programmers do not support the programming of the data flash. The programmers may be updated on request. This chapter is for information only.

56.7.1 Flash configuration in parallel Flash programming mode

Parallel Flash programming mode (MD[2:0] = 111):

MB91FV460B



Note: The address in parallel programming mode is listed here without 10:0000h offset.

Set the offset by keeping FA[22:20] = 001 the same kind as used for programming of the main flash.

Note: The “Dummy addresses for auto algorithm” are accepted although they are located below the physical addresses of the flash macro. This address space is needed to apply correct addresses in auto algorithms. See the example in [“Auto Program Algorithms” on page 1216](#).

56.7.2 Address mapping from CPU to parallel programming mode

The following tables show the calculation from CPU addresses to data flash macro addresses which are used in parallel programming.

● Address mapping MB91FV460B

CPU Address (addr)	Condition	Flash sectors	FA (flash address) Calculation
FFFB:FF00h to FFFC:FFFFh	-	SAS, SA0, SA1, SA2, SA3 (256 Byte + 64 Kbyte)	FA := addr - 0B:0000h

Note: FA result is without 10:0000h offset for parallel Flash programming.

Set the offset by keeping FA[22:20] = 001 the same kind as used for programming of the main flash.

56.7.3 Pin connections in parallel programming mode

Resetting after setting the MD[2:0] pins to [111] will halt CPU functioning. At this time, the Flash memory's interface circuit enables direct control of the Flash memory unit from external pins by directly linking some of the signals to General Purpose Ports. Please see table below for signal mapping.

In this mode, the Data Flash memory appears to the external pins as a stand-alone unit. This mode is generally set when writing/erasing using the parallel Flash programmer. In this mode, all operations of the Data Flash memory's Auto Algorithms are available.

Correspondence between flash macro and Flash Memory Control Signals

Data Flash macro pins	FR-CPU mode	MB91FV460B external pins			Comment
—	INITX	—	INITX		
FRSTX	—	FRSTX	P00_6		
—	—	MD_2	MD_2		Set to '1'
—	—	MD_1	MD_1		Set to '1'
—	—	MD_0	MD_0		Set to '1'
RDY	FMCS:RDY bit	RY/BYX	P00_0		
FCLK		FCLK	P00_2		Clock input
WEX	Internal control signal + control via interface circuit	WEX	P01_2		
OEX		OEX	P01_1		
CEX		CEX	P01_0		
RAS		RAS	P01_4		
EQIN		EQIN	P01_3		
LTIN		LTIN	P01_5		Set to '0'
—		TESTX	P00_3		Set to '1'
—		RDYI	P00_1		Set to '0'
FA0	Internal address bus	FA0	P05_7		
FA1 to FA8		FA1 to FA8	P07_0 to P07_7		
FA9 to FA16	Internal address bus	FA9 to FA16	P06_0 to P06_7		
—		FA17 to FA19	P05_0 to P05_2		Set to '0'
—		FA20 to FA22	P05_3 to P05_5		Set to "010"
DI0 to DI7, DO0 to DO7	Internal data bus	DQ0 to DQ7	P03_0 to P03_7		
—		DQ0 to DQ7	P02_0 to P02_7		DQ0 to DQ7 output in parallel to P03_0:7

56.7.4 Wait time before data flash access in parallel programming mode

After power-on or the end of a Setting Initialization Request (INITX), the internal data flash security module fetches the security information. The parallel programmer cannot access the flash until the security vector fetch is finished and has to wait for the following time:

- Min waittime after VDD5/VDD5R power on : 2.9 ms
- Min waittime after INITX rising : 1.0 ms

56.8. Data Flash Security

56.8.1 Data Flash Security Operation

The data flash security protects the flash against unauthorized read and write access.

- A **read** access to protected flash will return data=0x00 without notification. There is no flag indicating that the read access was masked by data flash security module.
- A **write** access to a write-protected sector will be cancelled. The flash macro will be put into RESET state, and the security macro will re-fetch the security information. It may take up to 600μs until the data flash can be accessed again. In direct access mode, the toggle bits will not change and the bit DFCS:RDY will not go to low state. In command sequencer mode, the flag DFWS:WERINT is set, indicating that the write operation was not successful.
- The only possible write operation to a protected sector is Chip Erase.
- The data flash security can be disabled by setting the external pin FSC_DISABLE = 1.
- After INIT, please wait 3 ms before accessing the data flash. This time is needed for the security vector fetch as well as internal signal synchronisation. This time is valid also if FSC_DISABLE = 1.

56.8.2 Security Vectors

Two 16-bit Data Flash Security Vectors (DFSV1, DFSV2) are located in the 256 byte security sector, controlling the protection functions of the Data Flash Security module:

DFSV1[15:0]: 0xFFFFB:FF00

DFSV2[15:0]: 0xFFFFB:FF02

Address	Vectors				Block
	+0	+1	+2	+3	
FFFBFF00 _H	DFSV1[15:0]		DFSV2[15:0]		Data Flash Security Vectors

56.8.3 Security Vector DFSV1 (bit15-bit0)

The setting of the Flash Security Vector DFSV1 is responsible for the read and write protection modes.

Explanation of the bits in the Flash Security Vector DFSV1 [15:0]

DFSV1[15:3]	DFSV1[2] Write Protection Level	DFSV1[1] Write Protection	DFSV1[0] Read Protection	Flash Security Mode
set all to "0"	set to "0"	set to "0"	set to "1"	Read Protection (all device modes, except INTVEC ¹)
set all to "0"	set to "0"	set to "1"	set to "0"	Write Protection (all device modes, without exception)
set all to "0"	set to "0"	set to "1"	set to "1"	Read Protection (all device modes, except INTVEC) and Write Protection (all device modes, without exception)
set all to "0"	set to "1"	set to "0"	set to "1"	Read Protection (all device modes, except INTVEC)
set all to "0"	set to "1"	set to "1"	set to "0"	Write Protection (all device modes, except INTVEC)
set all to "0"	set to "1"	set to "1"	set to "1"	Read Protection (all device modes, except INTVEC) and Write Protection (all device modes, except INTVEC)

1. INTVEC mode is the Internal Vector Fetch mode (MD[2:0] = "000")

Note : If Read Protection is set and the device is not in INTVEC mode and the data flash is written using the Command Sequencer write access, then the command sequencer will set the error flag because it cannot check that the flash programming was successful.

56.8.4 Security Vector DFSV2

The setting of the Flash Security Vector DFSV2 bits [15:0] is responsible for the individual write protection of the Data Flash sectors. It is only evaluated if write protection bit DFSV1 [1] is set.

Explanation of the bits in the Flash Security Vector DFSV2[15:0]

DFSV2 bit	Sector	Enable Write Protection	Disable Write Protection	Comment
DFSV2[0]	SA0	set to "0"	set to "1"	
DFSV2[1]	SA1	set to "0"	set to "1"	
DFSV2[2]	SA2	set to "0"	set to "1"	
DFSV2[3]	SA3	set to "0"	set to "1"	
DFSV2[7:4]	—	—	—	sectors not available
DFSV2[8]	SAS	set to "0"	—	write protection is mandatory!
DFSV2[15:9]	—	—	—	sectors not available

Note: It is mandatory to always set the sector where the Flash Security Vectors DFSV1 and DFSV2 are located to write protected (here sector SAS). Otherwise it is possible to overwrite the Security Vector to a setting where it is possible to either read out the Flash content or manipulate data by writing.

See section ["Data Flash access in CPU mode" on page 1215](#) for an overview about the sector organisation of the Flash Memory.

56.8.5 Data Flash Security Registers

The Data Flash Security module can be used to calculate a CRC over the Data Flash contents. And it is possible to force a security vector re-fetch by using the following registers.

● DFSCR0 : Data Flash Security Control Register 0

Address	31:24	23:16	15:8	7:0	bit
07118 _H	S[7:0] CRC[31:24]	CRC[23:0]			
	1111 1111	1111 1111	1111 1111	1111 1111	Initial value
	W, R	R	R	R	Attribute

S[7:0]	Sequence Activation
0xA5 --> 0x5A	Start of a Flash Security Vector Re-Fetch Sequence (write only)
0xF0 --> 0x0F	Start of a Flash Memory CRC32 Checksum Sequence (write only)

- Continuously writing "A5_H", "5A_H" in the DFSCR0[31:24] register will start a Flash Security Vector Re-fetch sequence immediately after writing "5A_H". There is no time restrictions between "A5_H" and "5A_H", but if "A5_H" is written followed by the one other than "5A_H", it must be written "A5_H" again. If not, the Re-Fetch sequence cannot be started even if "5A_H" is written.
- Continuously writing "F0_H", "0F_H" in the DFSCR0[31:24] register will start a CRC32 checksum sequence immediately after writing "0F_H". There is no time restrictions between "F0_H" and "0F_H", but if "F0_H" is written followed by the one other than "0F_H", it must be written "F0_H" again. If not, the CRC checksum sequence cannot be started even if "0F_H" is written.
- These bits are cleared by an INIT signal from external pin (INITX) or hardware watchdog and can be written

MB91460 Series

only.

Note: The Flash Security Vector Re-Fetch sequence is especially intended to be used after a chip erase command to update the security status without the need of applying an external INITX reset or after changing the status of the DFSV1 security vector.

Note: The CRC calculation runs on the internal RC clock. It is recommended to switch the RC clock frequency to 2 MHz for shortening the calculation time. However, the CPU clock (CLKB) must be faster than RC clock, otherwise the CRC calculation may not start correctly.

CRC[31:0]	CRC checksum result
	CRC checksum result (read only)

- This register contains the CRC32 checksum result after completion of the CRC32 checksum sequence (the sequence completion is indicated by DFSCR1.RDY). The CRC checksum is calculated in a standard CRC32/AAL5 algorithm with the polynomial $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$.
- These bits are set to 0xFFFFFFFF by an INIT signal from external pin (INITX) or hardware watchdog and can be read only.

● DFSCR1 : Data Flash Security Control Register 1

Address	31:24	23:16	15:8	7:0	bit
0711C _H	SVF_RDY--- RDY	---- CSZ[3:0]	CSA[15:0]		
	0xxx xxx1	0000 0000	0000 0000	0000 0000	Initial value
	R/Wx	R, R/W	R/W	R/W	Attribute

- Bit30-25: Reserved bits. The read value is always "X".
- Bit23-20: Reserved bits. The read value is always "0".

Bit 31:

SVF_RDY	Security Vector Fetch Ready (flag)
0	The security vector has not been fetched. The data flash is protected against read and write access. Read operations to data flash return 0x00. Write operations are ignored.
1	The security vector has been done. The data flash can be accessed according to the security settings.

Bit 24:

RDY	CRC Sequence Ready (flag)
0	CRC sequence running or not yet started
1	CRC sequence ready (data in the DFSCR0 register is valid)

Bit 19-16:

CSZ[3:0]	CRC Size Mask
0000	CRC size mask is 256 Byte
0001	CRC size mask is 512 Byte
0010	CRC size mask is 1 KByte

CSZ[3:0]	CRC Size Mask
0011	CRC size mask is 2 KByte
0100	CRC size mask is 4 KByte
0101	CRC size mask is 8 KByte
0110	CRC size mask is 16 KByte
0111	CRC size mask is 32 KByte
1000	CRC size mask is 64 KByte
1001 - 1111	Not supported

- CSZ3-0 is used as an OR-mask for the address given by CSA15-0. See address calculation below.
- These bits are cleared by an INIT signal from external pin (INITX) or hardware watchdog.

Bit 15-0

CSA[15:0]	CRC Calculation Start Address
0x00FF	CRC start address is 0x0FF00 (sector SAS start)
0x0100	CRC start address is 0x10000 (sector SA0 start)
0x0140	CRC start address is 0x14000 (sector SA1 start)

Notes: The values given above are just examples. The addresses to be written in this register are flash memory addresses like used in the flash parallel programming mode and not the mapped addresses which are used in CPU mode. See [“Address mapping from CPU to parallel programming mode” on page 1225](#).

- The CSA register contains the CRC start address which is aligned to 256 Byte addresses. It is only possible to calculate the CRC checksum over addresses located in the Data Flash Memory address space. Other addresses are invalid and might lead to wrong checksums.

■ Calculation of the CRC Start- and End-addresses

The CSZ3-0 setting is first translated into a mask value:

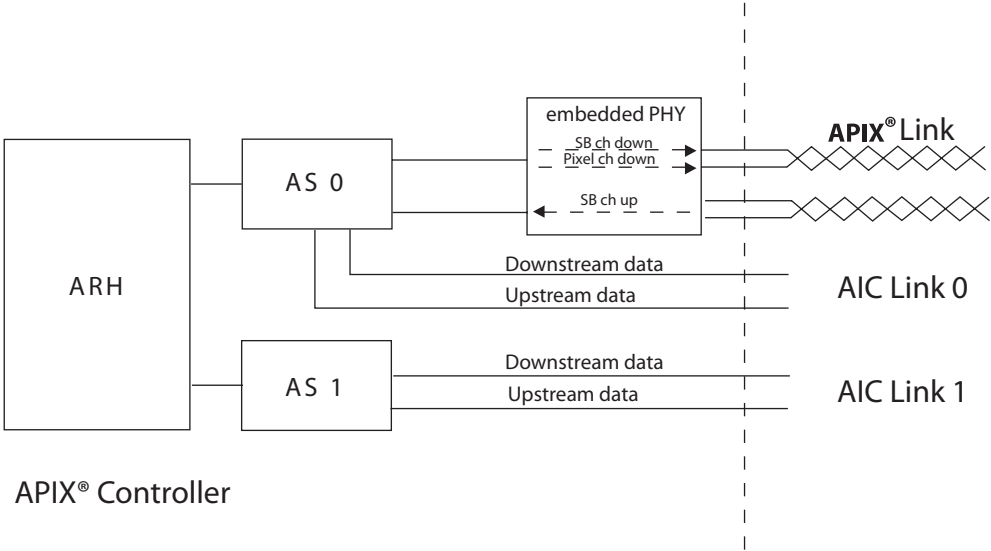
CSZ3-0	MASK
0000	0000_0000_0000_0000
0001	0000_0000_0000_0001
0010	0000_0000_0000_0011
0011	0000_0000_0000_0111
0100	0000_0000_0000_1111
0101	0000_0000_0001_1111
0110	0000_0000_0011_1111
0111	0000_0000_0111_1111
1000	0000_0000_1111_1111
1001-1111	and so on...

- CRC Start address = $CSA[15:0] \ll 8 + 0x00$
- CRC End address = $(CSA[15:0] \text{ or } MASK) \ll 8 + 0xFF$

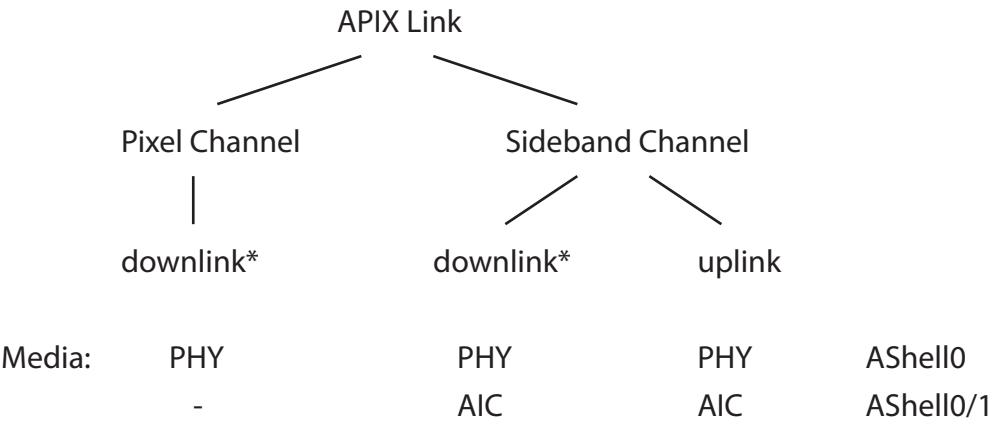
Chapter 57 APIX® Controller

57.1. Overview

The intergrated APIX® controller provides 2 links. Link 0 can be configured as an APIX® link or an Automotive Interconnect (AIC) link. Link 1 only supports AIC link.



*Remark: Link 1 can be used only if Link 0 is activated (CHCTRL.TXCFG = 0)



*Remark: MB91460S series provides either downlink over Pixelchannel or over Sidebandchannel

57.2. Automotive Remote Handler

The Automotive Remote Handler provides an Interface to the APIX® controller.

57.2.1 Register Description

■ General Control

- RHCTRL: Address 07200h

RHCTRL	31	30	29	28	27	26	25	24
	UNLOCK	CANCEL	-	-	TBNO[3]	TBNO[2]	TBNO[1]	TBNO[0]
	R0/W	R0/W	R0	R0	R/W	R/W	R/W	R/W
	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-
	R0	R0	R0	R0	R0	R0	R0	R0
	15	14	13	12	11	10	9	8
	WDG1	WDG0	FAT1	FAT0	-	LV	OFL	EV
	R	R	R	R	R0	R	R	R
	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-
	R0	R0	R0	R0	R0	R0	R0	R0

UNLOCK	0(def)	Transaction on buffer TBNO is requested
	1	Request unlock on waiting buffer TBNO
Caution: Requested data gets lost or data is being received after using this buffer with same IDX		
CANCEL	0(def)	Transaction on buffer TBNO is requested
	1	Request cancel on pending buffer TBNO
TBNO[3:0]	0-15	Writing starts transaction on buffer number TBNO
WDG1		readonly flag of enabled and selected CHWDG1.WDTXIRQx or enabled and selected CHWDG1.WDRXIRQx
WDG0		readonly flag of enabled and selected CHWDG0.WDTXIRQx or enabled and selected CHWDG0.WDRXIRQx
FAT0		readonly flag of enabled CHCTRL0.FATIRQ
FAT1		readonly flag of enabled CHCTRL1.FATIRQ
LV		readonly flag of enabled EVCTRL.LVIRQ
EV		readonly flag of enabled EVCTRL.EVIRQ
OFL		readonly flag of enabled EVCTRL.OFLIRQ

■ Channel Control and Status

- CHCTRL0 (Link 0): Address 07208h
- CHCTRL1 (Link 1): Address 07214h

CHCTRL0-1	31	30	29	28	27	26	25	24
	-	-	-	reserved	BYPASS	-	-	-
	R0/WX	R0/WX	R0/WX	R/W0	R/W	R/W	R/W	R/W
	23	22	21	20	19	18	17	16
	FATAL	UPHSK	DNHSK	-	-	-	FATIEN	FATIRQ
	R	R	R	RX/W	RX/W	RX/W	R/W	R(RM1)/W
	15	14	13	12	11	10	9	8
	reserved	UPRDY	CONNECTED	CRCERR	CRCTOUT	PERROR	READY	REMOTERST
	R	R	R	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W	R(RM1)/W
	7	6	5	4	3	2	1	0
	-	-	UPVALID	DNVALID	-	TXCFG	RSTRTA	INITRH
	RX/WX	RX/WX	R(W)	R(W)	R0	R/W	R/W	R/W

Bit28 reserved Bit

Always write 0 to this bit. The read value is the value written.

BYPASS: 0 Remote Handler active
1 Remote Handler inactive

In BYPASS mode Transaction Buffer 2 (for AShell 1) is used for downstream data (outbound) and Transaction Buffer 3 (for AShell 1) is used for upstream data (inbound).

Valid written data in Transaction Buffer 0/2 is delivered to AShell by setting DNVALID.

Valid received data in Transaction Buffer 1/3 from AShell is marked by setting UPVALID.

FATAL	1	indicates that AShell has encountered conditions where AShell can not continue to deliver and receive transactions. FATAL is only one CLKB cycle active.
UPHSK	1	indicates inbound handshake is performed
DNHSK	1	indicates outbound handshake is performed
FATIEN	0(def)	FATAL Interrupt disabled
	1	FATAL Interrupt enabled
FATIRQ	0(def)	FATAL Interrupt not active
	1	FATAL Interrupt active, triggered by FATAL

*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored

*Remark: While Fatal Interrupt is active, the corresponding channel is deactivated and the triggered buffers are canceled.

UPRDY	1	indicates that upstream serial channel (APIX® PHY) is operational
CONNECTED	1	a connection to remote APIX® is established
CRCERR	1	indicates occurrence of CRC error in upstream data (inbound)

*Remark: On a RMW instruction a '1' is read; write '0' clears the flag; write '1' is ignored

CRCTOUT	1	indicates occurrence of CRC timeout in upstream data (inbound)
---------	---	--

*Remark: On a RMW instruction a '1' is read; write '0' clears the flag; write '1' is ignored

PERROR	1	indicates occurrence of a protocol error
--------	---	--

*Remark: On a RMW instruction a '1' is read; write '0' clears the flag; write '1' is ignored

READY 1 indicates that AShell is ready to accept outbound transactions

REMODERST 1 indicates a restart of remote AShell was performed

*Remark: On a RMW instruction a '1' is read; write '0' clears the flag; write '1' is ignored

UPVALID BYPASS==0 read only Read only status (ap_data_out_valid)

BYPASS==1 0(def) Cleared by SW after successful reception (read) of upstream data
1 Set by HW to mark upstream data as valid (ap_data_out_valid)

*Remark: Reading UPVALID returns the status and then clears the flag value to '0'.

DNVALID BYPASS==0 read only DNVALID is only operational in BYPASS mode (always read 0)

BYPASS==1 0(def) Cleared by HW after successful transfer to AShell
1 Set by SW to mark downstream data as valid (ap_data_in_valid)

TXCFG 0 AShell and PHY running (write protection on APCFG registers)

1(def) AShell and PHY configuration (possible to change APCFG registers)

RSTRTA 0 AShell running Level

1(def) AShell initialisation

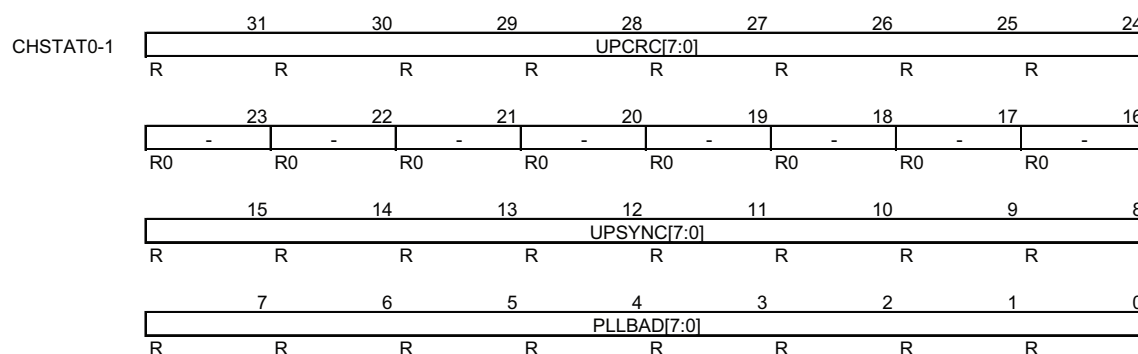
INITRH 0 Remote Handler running Level

1(def) Remote Handler initialisation (no change of TB* and TF* registers)

*Remark: PENDING requests (set while INIT==1) will be started with INIT==0

- CHSTAT0 (Link 0): Address 0720Ch

- CHSTAT1 (Link 1): Address 07218h



UP_CRC 0-255 Inbound CRC errors

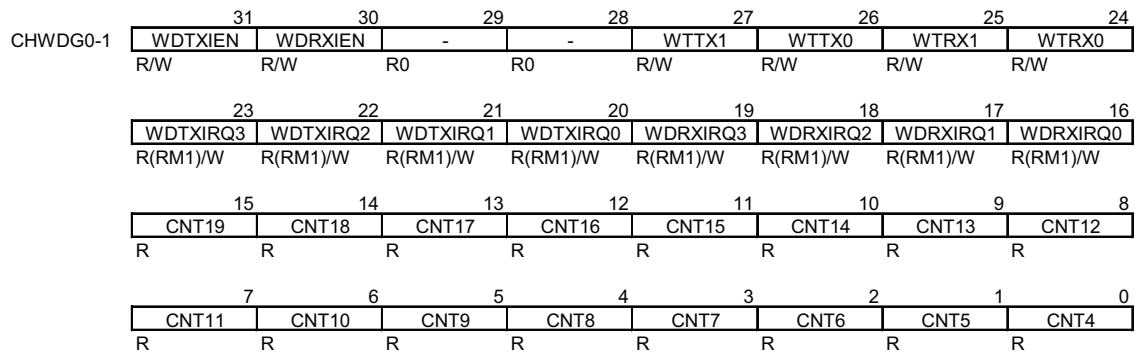
UP_SYNC 0-255 Synchronisation losses

PLL_BAD 0-255 PLL synchronisation losses

■ Channel Watchdog

MB91460 Series

- CHWDG0 (Link 0): Address 07210h
- CHWDG1 (Link 1): Address 0721Ch

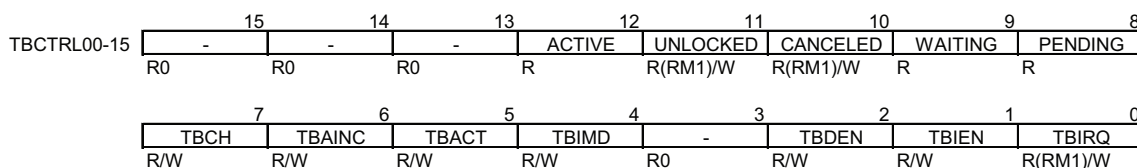


WDTXIEN	0(def)	Watchdog interrupt for TX is disabled
	1	Watchdog interrupt for TX is enabled
WDRXIEN	0(def)	Watchdog interrupt for RX is disabled
	1	Watchdog interrupt for RX is enabled
WTTX	0	select WDTXIRQ0
	1	select WDTXIRQ1
	2	select WDTXIRQ2
	3	select WDTXIRQ3
WTRX	0	select WDRXIRQ0
	1	select WDRXIRQ1
	2	select WDRXIRQ2
	3	select WDRXIRQ3
WDTXIRQ3	0(def)	interrupt for TX at 2^{19} is not active
	1	interrupt for TX at 2^{19} is active
WDTXIRQ2	0(def)	interrupt for TX at 2^{16} is not active
	1	interrupt for TX at 2^{16} is active
WDTXIRQ1	0(def)	interrupt for TX at 2^{14} is not active
	1	interrupt for TX at 2^{14} is active
WDTXIRQ0	0(def)	interrupt for TX at 2^{13} is not active
	1	interrupt for TX at 2^{13} is active
WDRXIRQ3	0(def)	interrupt for RX at 2^{19} is not active
	1	interrupt for RX at 2^{19} is active
WDRXIRQ2	0(def)	interrupt for RX at 2^{18} is not active
	1	interrupt for RX at 2^{18} is active
WDRXIRQ1	0(def)	interrupt for RX at 2^{17} is not active
	1	interrupt for RX at 2^{17} is active
WDRXIRQ0	0(def)	interrupt for RX at 2^{16} is not active
	1	interrupt for RX at 2^{16} is active

*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored
CNT upper 16 Bit of the 20 Bit watchdog freerun timer

Transaction Buffer Control

- TBCTRL00 (Transaction Buffer 00): Address 07220h
- TBCTRL01 (Transaction Buffer 01): Address 07222h
- TBCTRL02 (Transaction Buffer 02): Address 07224h
- TBCTRL03 (Transaction Buffer 03): Address 07226h
- TBCTRL04 (Transaction Buffer 04): Address 07228h
- TBCTRL05 (Transaction Buffer 05): Address 0722Ah
- TBCTRL06 (Transaction Buffer 06): Address 0722Ch
- TBCTRL07 (Transaction Buffer 07): Address 0723Eh
- TBCTRL08 (Transaction Buffer 08): Address 07230h
- TBCTRL09 (Transaction Buffer 09): Address 07232h
- TBCTRL10 (Transaction Buffer 10): Address 07234h
- TBCTRL11 (Transaction Buffer 11): Address 07236h
- TBCTRL12 (Transaction Buffer 12): Address 07238h
- TBCTRL13 (Transaction Buffer 13): Address 0723Ah
- TBCTRL14 (Transaction Buffer 14): Address 0723Ch
- TBCTRL15 (Transaction Buffer 15): Address 0723Eh



ACTIVE	0	No active data in Transaction Buffer
	1	Active data in Transaction Buffer (delivery to AShell requested)
UNLOCKED	0(def)	No last action on this buffer
	1	Last action on this Transaction Buffer was UNLOCK

*Remark: On a RMW instruction a '1' is read; write '0' clears the register write '1' is ignored

CANCELED	0(def)	No last action on this buffer
	1	Last action on this Transaction Buffer was a succesful CANCEL

*Remark: On a RMW instruction a '1' is read; write '0' clears the register write '1' is ignored

WAITING	0	Not waiting for requested data
	1	Transaction Buffer waiting for requested data

*Remark: WAITING will be cleared at reception of requested data in buffer

PENDING	0	No pending data in Transaction Buffer
	1	Pending data in Transaction Buffer (not yet requested delivery to AShell)

TBCH	0(def)	Transaction Buffer assigned to channel 0
	1	Transaction Buffer assigned to channel 1

MB91460 Series

TBAINC	0(def)	Transaction Buffer Address increment disabled
	1	Transaction Buffer Address increment enabled
		1. increments address after WR access to TBDATA and transmission of TF
		2. increments address after reception of requested TF and RD access to TBDATA, then autonomous transmission of next read request

Remark: The address increment depends on the setting of TFCTRL.SIZE

TFCTRL.SIZE	00	increment by 1
TFCTRL.SIZE	01	increment by 2
TFCTRL.SIZE	10	increment by 4

*Remark: Address increment is only supported if TBACT = 1.

TBACT	0(def)	Transaction Buffer will be activated by WR access to TBNO
	1	Transaction Buffer will be activated by WR access to TBNO or TFDATA (RD and WR)
TBIMD	0(def)	Transaction Buffer Interrupt on TB idle (after transaction send)
	1	Transaction Buffer Interrupt on TB valid (after read request data reception)
TBIEN	0(def)	Transaction Buffer Interrupt disabled
	1	Transaction Buffer Interrupt enabled
TBDEN	0(def)	Transaction Buffer DMA disabled
	1	Transaction Buffer DMA enabled
TBIRQ	0	Transaction Buffer Interrupt not active
	1	Transaction Buffer Interrupt active

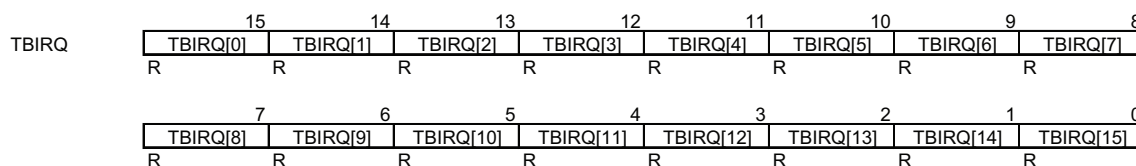
*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored

*Remark: TBIRQ can/will be cleared by the following events:

1. Cleared by SW on write access to TBIRQ flag with data '0'
2. Cleared by HW if TBACT==1 and read or write access to TBDATA register (both CPU or DMA)
3. Cleared by HW if TBACT==1 and DMA asserts hardware clear signal IIOC

Transaction Buffer Interrupt

- TBIRQ: Address 07240h



TBIRQ[15:0] Read only flags of enabled (TBIEN==1) TBCTRLxx.TBIRQ

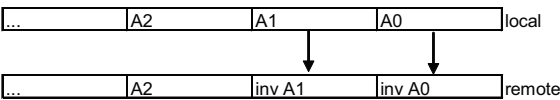
Transaction Frame

- TFCTRL00 (Transaction Buffer 00): Address 07250h
- TFCTRL01 (Transaction Buffer 01): Address 07252h
- TFCTRL02 (Transaction Buffer 02): Address 07254h
- TFCTRL03 (Transaction Buffer 03): Address 07256h

- TFCTRL04 (Transaction Buffer 04): Address 07258h
- TFCTRL05 (Transaction Buffer 05): Address 0725Ah
- TFCTRL06 (Transaction Buffer 06): Address 0725Ch
- TFCTRL07 (Transaction Buffer 07): Address 0725Eh
- TFCTRL08 (Transaction Buffer 08): Address 07260h
- TFCTRL09 (Transaction Buffer 09): Address 07262h
- TFCTRL10 (Transaction Buffer 10): Address 07264h
- TFCTRL11 (Transaction Buffer 11): Address 07266h
- TFCTRL12 (Transaction Buffer 12): Address 07268h
- TFCTRL13 (Transaction Buffer 13): Address 0726Ah
- TFCTRL14 (Transaction Buffer 14): Address 0726Ch
- TFCTRL15 (Transaction Buffer 15): Address 0726Eh

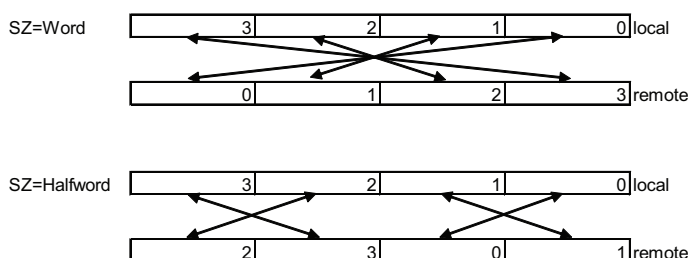
TFCTRL00-15			7	6	5	4	3	2	1	0
			TFDSWP	TFAINV	-	ERROR	SZ[1]	SZ[0]	OAEN	RW
			R/W	R/W	R0	R	R/W	R/W	R/W	R/W
RW	0	Read								
	1	Write								
OAEN	0	Offset address disabled								
	1	Offset address enabled								
SZ[1:0]	00	Byte								
	01	Halfword								
	10	Word								
	11	-								
ERROR	0	Normal operation								
	1	Remote Handler RX bus error occurred								
TFAINV	0	Normal mode								
	1	Address inversion								

In address inversion mode the two least significant bits of the address are inverted

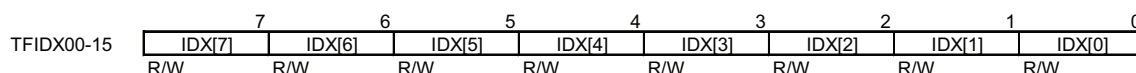


TFDSWP	0	Normal mode
	1	Byte swapping

In swapping mode depending on the configured size the following byte swapping of the data is performed



- TFIDX00 (Transaction Buffer 00): Address 07251h
- TFIDX01 (Transaction Buffer 01): Address 07253h
- TFIDX02 (Transaction Buffer 02): Address 07255h
- TFIDX03 (Transaction Buffer 03): Address 07257h
- TFIDX04 (Transaction Buffer 04): Address 07259h
- TFIDX05 (Transaction Buffer 05): Address 0725Bh
- TFIDX06 (Transaction Buffer 06): Address 0725Dh
- TFIDX07 (Transaction Buffer 07): Address 0725Fh
- TFIDX08 (Transaction Buffer 08): Address 07261h
- TFIDX09 (Transaction Buffer 09): Address 07263h
- TFIDX10 (Transaction Buffer 10): Address 07265h
- TFIDX11 (Transaction Buffer 11): Address 07267h
- TFIDX12 (Transaction Buffer 12): Address 07269h
- TFIDX13 (Transaction Buffer 13): Address 0726Bh
- TFIDX14 (Transaction Buffer 14): Address 0726Dh
- TFIDX15 (Transaction Buffer 15): Address 0726Fh



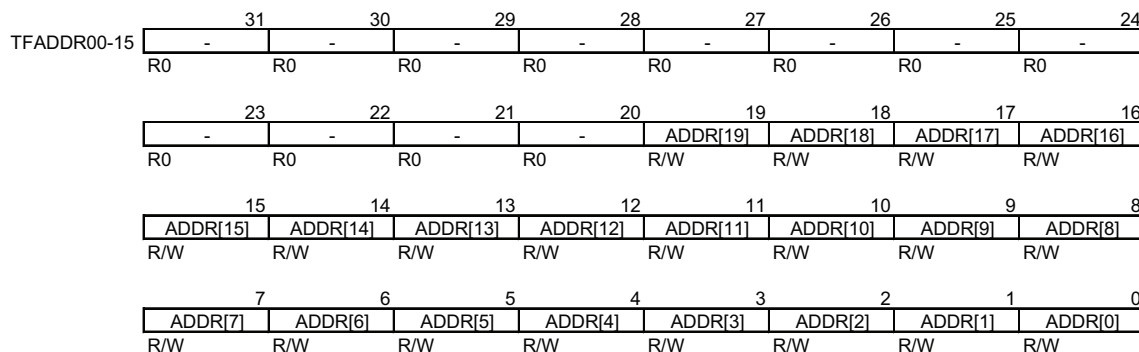
IDX[7:0] Any number between 0 and 255

*Remark: Index is used for read request. Received data from a read request will be stored in an active Transaction Buffer with matching index.

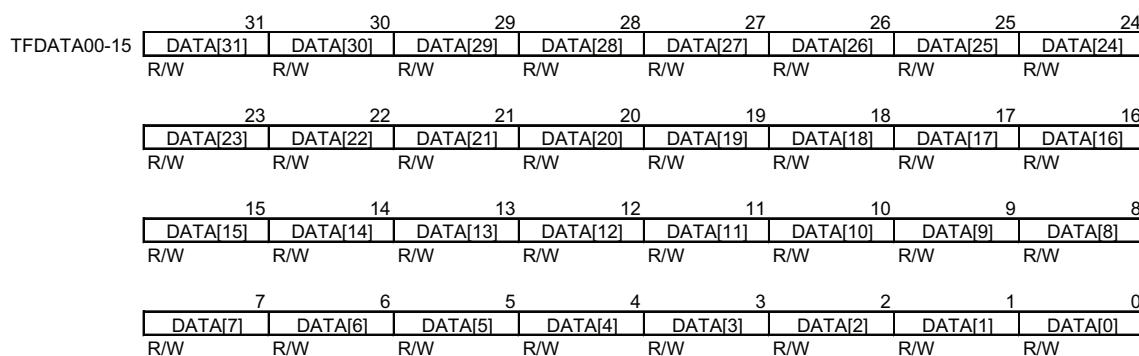
If there is no active Transaction Buffer with matching index (e.g. by using UNLOCK), the received data is discarded.

- TFADDR00 (Transaction Buffer 00): Address 07270h
- TFADDR01 (Transaction Buffer 01): Address 07274h
- TFADDR02 (Transaction Buffer 02): Address 07278h
- TFADDR03 (Transaction Buffer 03): Address 0727Ch
- TFADDR04 (Transaction Buffer 04): Address 07280h
- TFADDR05 (Transaction Buffer 05): Address 07284h
- TFADDR06 (Transaction Buffer 06): Address 07288h
- TFADDR07 (Transaction Buffer 07): Address 0728Ch
- TFADDR08 (Transaction Buffer 08): Address 07290h
- TFADDR09 (Transaction Buffer 09): Address 07294h
- TFADDR10 (Transaction Buffer 10): Address 07298h
- TFADDR11 (Transaction Buffer 11): Address 0729Ch
- TFADDR12 (Transaction Buffer 12): Address 072A0h

- TFADDR13 (Transaction Buffer 13): Address 072A4h
- TFADDR14 (Transaction Buffer 14): Address 072A8h
- TFADDR15 (Transaction Buffer 15): Address 072ACH



- TFDATA00 (Transaction Buffer 00): Address 072B0h
- TFDATA01 (Transaction Buffer 01): Address 072B4h
- TFDATA02 (Transaction Buffer 02): Address 072B8h
- TFDATA03 (Transaction Buffer 03): Address 072BCh
- TFDATA04 (Transaction Buffer 04): Address 072C0h
- TFDATA05 (Transaction Buffer 05): Address 072C4h
- TFDATA06 (Transaction Buffer 06): Address 072C8h
- TFDATA07 (Transaction Buffer 07): Address 072CCh
- TFDATA08 (Transaction Buffer 08): Address 072D0h
- TFDATA09 (Transaction Buffer 09): Address 072D4h
- TFDATA10 (Transaction Buffer 10): Address 072D8h
- TFDATA11 (Transaction Buffer 11): Address 072DCh
- TFDATA12 (Transaction Buffer 12): Address 072E0h
- TFDATA13 (Transaction Buffer 13): Address 072E4h
- TFDATA14 (Transaction Buffer 14): Address 072E8h
- TFDATA15 (Transaction Buffer 15): Address 072ECh

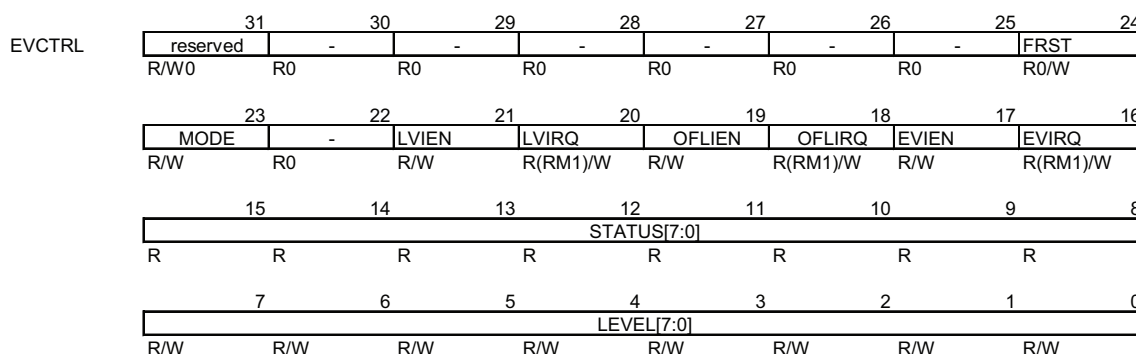


ADDR[19:0] Address in remote system

DATA[31:0] Payload data

■ Event Control

- EVCTRL: Address 072F0h



Bit31 reserved Bit

Always write 0 to this bit. The read value is the value written.

FRST	0(def)	FIFO in normal operation
	1	FIFO pointers are reset pulse (set to 0 after 1 cycle)
MODE	0 (def)	level mode On full FIFO new Events are discarded
	1	ring mode
LVLEN	0(def)	Level Interrupt disabled
	1	Level Interrupt enabled
LVIRQ	0(def)	Level Interrupt not active
	1	Level Interrupt active (if STATUS>=LEVEL)

*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored

OFLLEN	0(def)	Event Buffer Overflow Interrupt disabled
	1	Event Buffer Overflow Interrupt enabled
OFLIRQ	0(def)	Event Buffer Overflow Interrupt not active
	1	Event Buffer Overflow Interrupt active

*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored

EVLEN	0(def)	Event Buffer Interrupt disabled
	1	Event Buffer Interrupt enabled
EVIRQ	0(def)	Event Buffer Interrupt not active
	1	Event Buffer Interrupt active

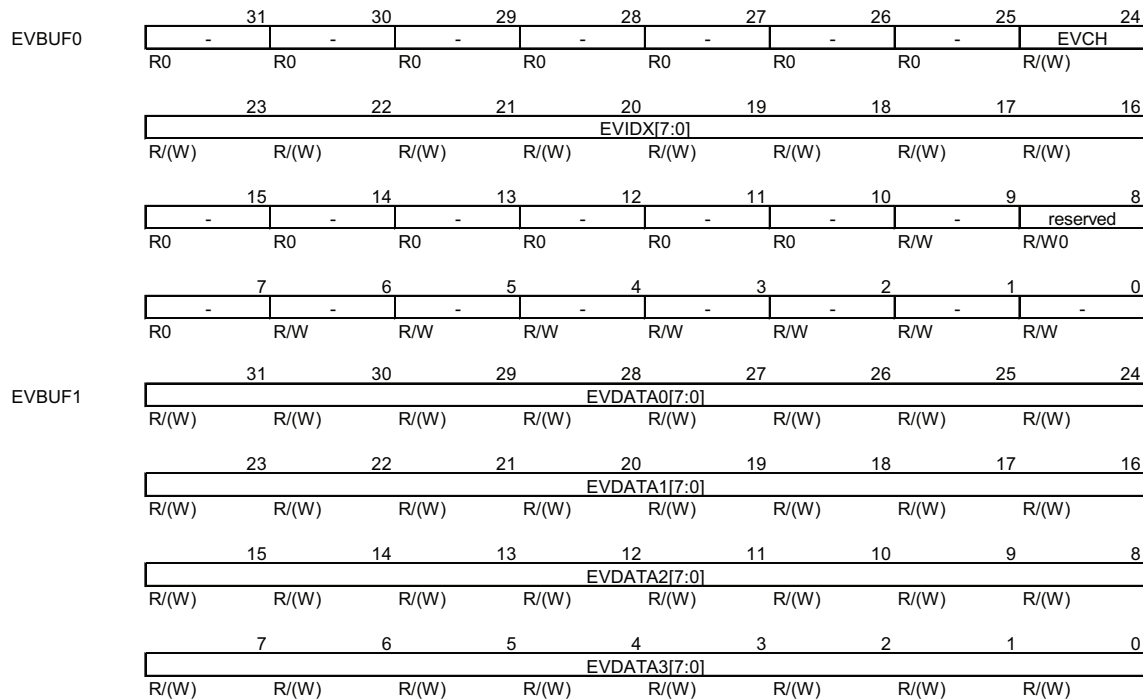
*Remark: On a RMW instruction a '1' is read; write '0' clears the interrupt; write '1' is ignored

Set by hardware, reset by software

STATUS[7:0]	0-128	Current FIFO filling status read only
LEVEL[7:0]	0-128	FIFO interrupt level (128 default)

■ Eventbuffer

- EVBUF0: Address 072F8h
- EVBUF1: Address 072FCh



EVCH 0-1 Holds channel number from Remote Handler RX event

EVIDX[7:0] 0-255 Holds index number from Remote Handler RX event

Bit8 reserved Bit

Always write 0 to this bit. The read value is the value written.

EVDATA0-3 4 bytes of payload data

*Remark: It is recommended to read first EVBUF0 and after that EVBUF1.

A read access to EVBUF0 triggers a retrieve of the current event message from the event buffer fifo and returns the channel number and event index.

A read access to EVBUF1 returns the data part of the a event message

■ Apix® configuration

- APCFG0x (Link 0)
- APXFG1x (Link 1)

APCFG00 07300h APCFG10 07310h	31	30	29	28	27	26	25	24
	config byte 1							
	R/W	0	R/W	0	R/W	0	R/W	0
	23	22	21	20	19	18	17	16
APCFG01 07304h APCFG11 07314h	config byte 2							
	R/W	0	R/W	0	R/W	1	R/W	0
	15	14	13	12	11	10	9	8
	config byte 3							
APCFG02 07308h APCFG12 07318h	R/W	0	R/W	0	R/W	0	R/W	0
	7	6	5	4	3	2	1	0
	config byte 4							
	R/W	1	R/W	0	R/W	0	R/W	0
APCFG03 0730Ch APCFG13 0731Ch	31	30	29	28	27	26	25	24
	config byte 5							
	R/W	1	R/W	1	R/W	1	R/W	0
	23	22	21	20	19	18	17	16
APCFG04 07308h APCFG14 07318h	config byte 6							
	R/W	0	R/W	0	R/W	0	R/W	0
	15	14	13	12	11	10	9	8
	config byte 7							
APCFG05 0730Ch APCFG15 0731Ch	R/W	0	R/W	0	R/W	0	R/W	0
	7	6	5	4	3	2	1	0
	config byte 8							
	R/W	0	R/W	1	R/W	0	R/W	0
APCFG06 07308h APCFG16 07318h	31	30	29	28	27	26	25	24
	config byte 9							
	R/W	0	R/W	0	R/W	0	R/W	1
	23	22	21	20	19	18	17	16
APCFG07 0730Ch APCFG17 0731Ch	config byte 10							
	R/W	0	R/W	0	R/W	0	R/W	1
	15	14	13	12	11	10	9	8
	config byte 11							
APCFG08 07308h APCFG18 07318h	R/W	0	R/W	1	R/W	0	R/W	0
	7	6	5	4	3	2	1	0
	config byte shell 1							
	R/W	0	R/W	0	R/W	1	R/W	1
APCFG09 0730Ch APCFG19 0731Ch	31	30	29	28	27	26	25	24
	config byte shell 2							
	R/W	1	R/W	0	R/W	0	R/W	0
	23	22	21	20	19	18	17	16
APCFG10 0730Ch APCFG20 0731Ch	config byte shell 3							
	R/W	1	R/W	0	R/W	0	R/W	1
	15	14	13	12	11	10	9	8
	config byte shell 4							
APCFG11 0730Ch APCFG21 0731Ch	R/W	0	R/W	0	R/W	0	R/W	0
	7	6	5	4	3	2	1	0
	config byte shell 5							
	R/W	0	R/W	0	R/W	0	R/W	0

AShell and PHY configuration.

■ Module ID

- MODULEID: Address 07320h
MODULEID[31:0]: Version of the APIX® controller

57.3. APIX® PHY Configuration

57.3.1 Powerdown

Configuration Vector:		APCFG 00
Bit	Default	Description
31	0	global power down (upstream, downstream and PLL) 1: power down 0: power up
29	0	power down serializer and output driver (diff amp) 1: power down 0: power up
28	0	power down upstream path 1: power down 0: power up

57.3.2 Nominal Current

Configuration Vector:		APCFG 01
Bit	Default	Description
19	0	nominal current setting (64 steps) 000000: min (0 mA - power down output driver) 111111: max
18	0	
17	0	
16	0	
15	0	
14	0	

MB91460 Series

57.3.3 Pre-emphasis

Configuration Vector:		APCFG 00
Bit	Default	Description
26	0	pre-emphasis configuration: reduce output current (pre-emphasis) after N equal serial bits (N = 0..7)
25	0	
24	0	
Configuration Vector:		APCFG 01
Bit	Default	Description
13	0	pre-emphasis current setting (64 steps) 000000: min (0 mA - power down output driver) 111111: max
12	0	
11	0	
10	0	
9	0	
8	0	

57.3.4 Sampling Offset

Configuration Vector:		APCFG 00
Bit	Default	Description
11	0	upstream sampling point configuration 0000: optimum sampling point when operating in 62.50 Mbit/s mode 0010: optimum sampling point when operating in 41.67 Mbit/s or 31.25 Mbit/s mode 0100: optimum sampling point when operating in 20.83 Mbit/s mode
10	0	
9	0	
8	0	

57.3.5 Charge Pump Control

Configuration Vector:		APCFG 01
Bit	Default *1	Description
23	1	charge pump current control
22	0	
21	0	
20	0	

1. Recommended settings: 1000 and higher.

57.4. DMA transfer request

To request a DMA transfer by a Transaction Buffer, please configure the transfer request source in DMACAx as follows.

IS	EIS(DDNO)	RN	Function	Transfer stop request
10000	1010	160	APIX® Transaction Buffer 0	available
10001	1010	161	APIX® Transaction Buffer 1	available
10010	1010	162	APIX® Transaction Buffer 2	available
10011	1010	163	APIX® Transaction Buffer 3	available
10100	1010	164	APIX® Transaction Buffer 4	available
10101	1010	165	APIX® Transaction Buffer 5	available
10110	1010	166	APIX® Transaction Buffer 6	available
10111	1010	167	APIX® Transaction Buffer 7	available
11000	1010	168	APIX® Transaction Buffer 8	available
11001	1010	169	APIX® Transaction Buffer 9	available
11010	1010	170	APIX® Transaction Buffer 10	available
11011	1010	171	APIX® Transaction Buffer 11	available
11100	1010	172	APIX® Transaction Buffer 12	available
11101	1010	173	APIX® Transaction Buffer 13	available
11110	1010	174	APIX® Transaction Buffer 14	available
11111	1010	175	APIX® Transaction Buffer 15	available

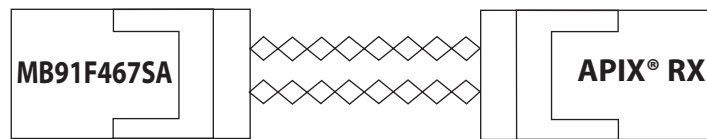
57.5. Automotive Interconnect Pins

The AIC Pins also serve as general ports.

Pin name	Pin function	I/O format	Pull-up Pull-down	Standby control	Setting required to use
RCK0	AIC uplink clock of Apix [®] link0	CMOS output and CMOS hysteresis, CMOS Automotive hysteresis, TTL input	Programmable	Provided	Set port function mode PFR28: Bit0 = 1, EPFR28: Bit0 = 1
RDA00	AIC uplink data of Apix [®] link0				Set port function mode PFR28: Bit1 = 1, EPFR28: Bit1 = 1
RDA01					Set port function mode PFR28: Bit2 = 1, EPFR28: Bit2 = 1
TDA00	AIC downlink data of Apix [®] link0				Set port function mode PFR28: Bit5 = 1, EPFR28: Bit5 = 1
TDA01					Set port function mode PFR28: Bit6 = 1, EPFR28: Bit6 = 1
TCLI0	AIC downlink clock of Apix [®] link0				Set port function mode PFR28: Bit7 = 1, EPFR28: Bit7 = 1
RCK1	AIC uplink clock of Apix [®] link1				Set port function mode PFR17: Bit0 = 1, EPFR17: Bit0 = 1
RDA10	AIC uplink data of Apix [®] link1				Set port function mode PFR17: Bit1 = 1, EPFR17: Bit1 = 1
RDA11					Set port function mode PFR17: Bit2 = 1, EPFR17: Bit2 = 1
TDA10	AIC downlink data of Apix [®] link1				Set port function mode PFR17: Bit5 = 1, EPFR17: Bit5 = 1
TDA11					Set port function mode PFR17: Bit6 = 1, EPFR17: Bit6 = 1
TCLI1	AIC downlink clock of Apix [®] link1				Set port function mode PFR17: Bit7 = 1, EPFR17: Bit7 = 1

57.6. USECASES

57.6.1 Communication over APIX® link



■ Downlink over Pixelchannel

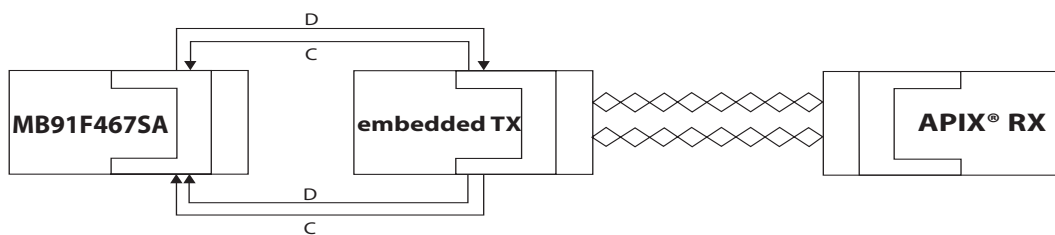
Downlink over Pixelchannel is provided by default configuration. Please configure the PHY according to Chapter “APIX ® PHY Configuration” on page1244

■ Downlink over Sidebandchannel

Register	Bit	Default	Value	Description
APCFG01	31	1	0	0: disable data mode / enable pixel stream mode 1: enable data mode / disable pixel stream mode

57.6.2 Communication over Automotive Interconnect to external AShell

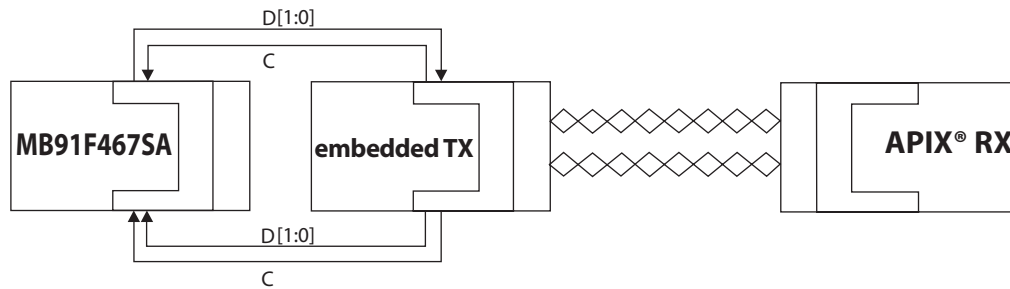
■ 1Bit Datawidth



Register	Bit	Default	Value	Description
APCFGn1	31	1	0	0: disable data mode / enable pixel stream mode 1: enable data mode / disable pixel stream mode
APCFGn1	29	1	0	1: enable core clock of APIX® PHY 0: disable
APCFGn3	23	1	0	1: sbup_data[1:0] 0: sbup_data[0]
APCFGn3	21	1	0	1: sbdown_data[1:0] 0: sbdown_data[0]
APCFGn3	18	0	1	AShell: connect internal Ashell to external APIX® PHY through GPIO interface 1: enable 0: disable

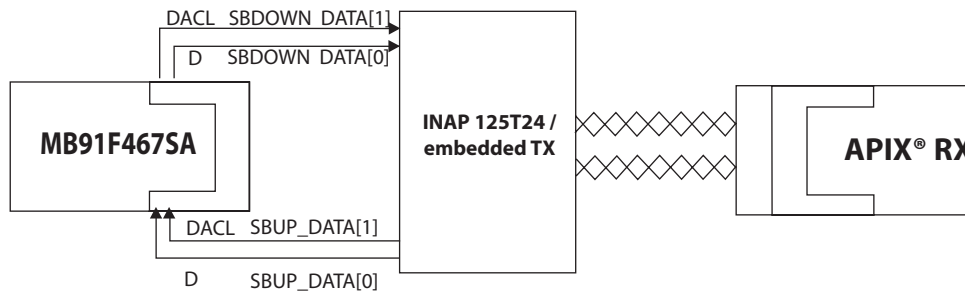
MB91460 Series

■ 2Bit Datawidth



Register	Bit	Default	Value	Description
APCFGn1	31	1	0	0: disable data mode / enable pixel stream mode 1: enable data mode / disable pixel stream mode
APCFGn1	29	1	0	1: enable core clock of APIX® PHY 0: disable
APCFGn3	18	0	1	AShell: connect internal Ashell to external APIX® PHY through GPIO interface 1: enable 0: disable

57.6.3 Communication over Automotive Interconnect to external PHY



Register	Bit	Default	Value	Description
APCFGn1	31	1	0	0: disable data mode / enable pixel stream mode 1: enable data mode / disable pixel stream mode
APCFGn1	29	1	0	1: enable core clock of APIX® PHY 0: disable
APCFGn3	22	0	1	AShell: validate sbup_data with 1: sbup_data[1] 0: sbup_valid

Register	Bit	Default	Value	Description
APCFGn3	20	0	1	AShell: generate sbdown clock and transmit as sbdown_data[1]
APCFGn3	19	0	0	11: disable 10: with use of internal counter (asynchronous to core_clk of APIX® PHY) 01: with use of sbdown_trigger (synchronous to core_clk of APIX® PHY) 00: disable
APCFGn3	18	0	1	AShell: connect internal Ashell to external APIX® PHY through GPIO interface 1: enable 0: disable
APCFGn3	2	0	t.b.d.	AShell: configures cycle time of sbdown clock (multiples of Ashell core clock) when sbdown_data are asynchronous (sbdown_data[1] is used as sbdown clock) or cfg_spi_over_sb is enabled 0x0B: recommended minimum (no low bandwidth mode, AShell and APIX® PHY operate at same core clock frequency) 0x14: recommended minimum (low bandwidth mode 2, AShell and APIX® PHY operate at 62.5 MHz) 0x26: recommended minimum (low bandwidth mode 1, AShell and APIX® PHY operate at 62.5 MHz)
	1	0		
	0	0		
	30	0		
	29	1		
	28	0		
	27	0		
	26	1		
	25	1		
	24	0		

57.6.4 Caution

Up to now only the usecases [“Communication over APIX® link” on page 1248](#) and [“Communication over Automotive Interconnect to external PHY” on page 1249](#) are guaranteed.

Chapter 58 E-Ray

58.1. About this Document

58.1.1 Conventions

The following conventions are used within this chapter:

CAPITALS: POC states and CHI commands

58.1.2 Definitions

FlexRay Frame: Header Segment + Payload Segment

Message Buffer: Header Section + Data Section

Message RAM: Header Partition + Data Partition

Data Frame: FlexRay frame that is not a null frame

58.1.3 Scope

This document describes the E-Ray FlexRay IP-module and its features from the application programmer's point of view.

58.1.4 Terms and Abbreviations

This document uses the following terms and abbreviations:

Term	Meaning
AP	Action Point
BD	Bus Driver
BSS	Byte Start Sequence
CAS	Collision Avoidance Symbol
CC	Communication Controller
CHI	Controller Host Interface
CIF	Customer Interface Block
CRC	Cyclic Redundancy Check
FES	Frame End Sequence
FSS	Frame Start Sequence
FIFO	First In First Out (message buffer structure)
FSM	Finite State Machine
FSP	Frame and Symbol Processing Block
FTM	Fault Tolerant Midpoint
GIF	Generic Interface Block
GTU	Global Time Unit Block
IBF	Input Buffer
INT	Interrupt Control Block
MHD	Message Handler Block
MT	Macrotick
MTS	Media Access Test Symbol
NCT	Network Communication Time
NEM	Network Management Block
NIT	Network Idle Time
NM	Network Management
OBF	Output Buffer
POC	Protocol Operation Control
PRT	Protocol Controller Block
SDL	Specification and Description Language
SUC	System Universal Control Block
TBF	Transient Buffer
TDMA	Time Division Multiple Access (media access method)
TSS	Transmission Start Sequence
TT-D	Time Triggered Distributed Synchronization
μT	Microtick
WUP	Wakeup Pattern
WUS	Wakeup Symbol

MB91460 Series

58.2. Overview

The E-Ray IP-module performs communication according to the FlexRay protocol specification v2.1. With maximum specified sample clock the bitrate is 10 MBit/s.

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message storage consists of a single-ported Message RAM that holds up to 128 message buffers. All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the E-Ray IP-module can be accessed directly by an external Host via the module's Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, and to access the Message RAM via Input / Output Buffer.

The E-Ray IP-module supports the following features:

- Conformance with FlexRay protocol specification v2.1
- Data rates of up to 10 Mbit/s on each channel
- Up to 128 message buffers configurable
- 8 Kbyte of Message RAM for storage of e.g. 128 message buffers with max. 48 byte data section or up to 30 message buffers with 254 byte data section
- Configuration of message buffers with different payload lengths possible
- One configurable receive FIFO
- Each message buffer can be configured as receive buffer, as transmit buffer or as part of the receive FIFO
- Host access to message buffers via Input and Output Buffer
Input Buffer: Holds message to be transferred to the Message RAM
Output Buffer: Holds message read from the Message RAM
- Filtering for slot counter, cycle counter, and channel
- Maskable module interrupts
- Network Management supported

58.2.1 Block Diagram

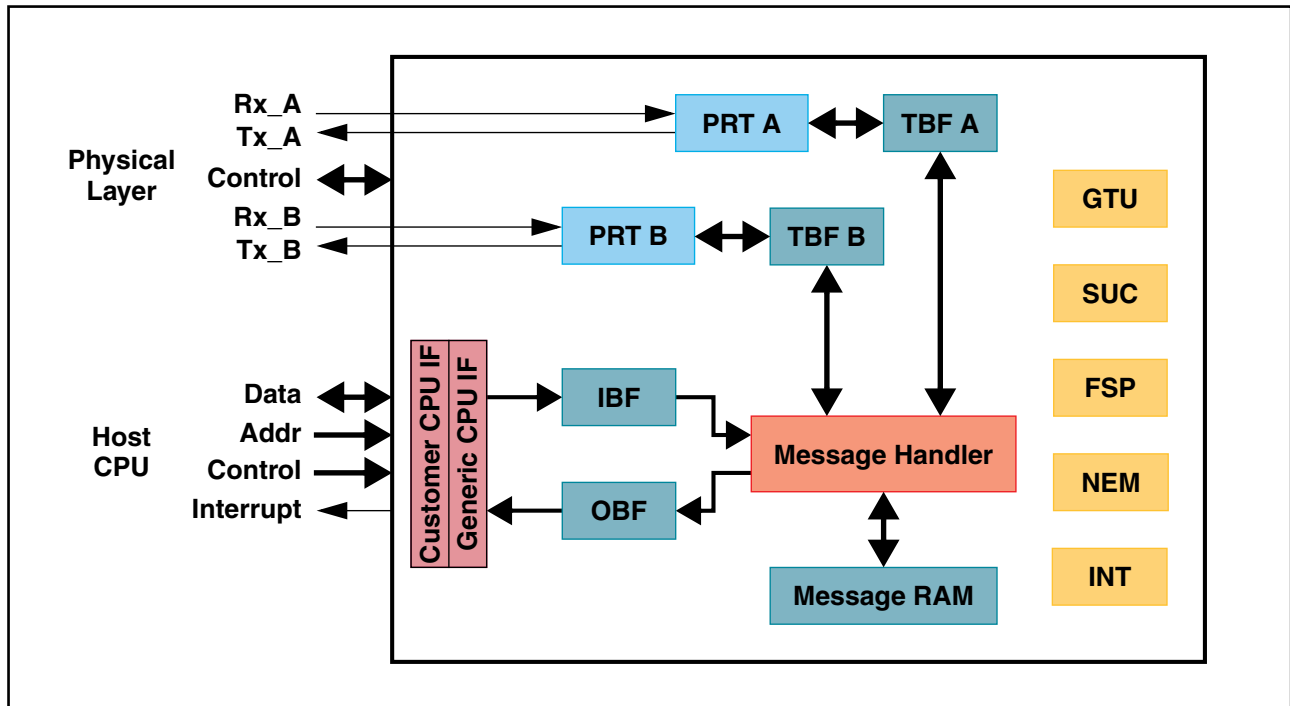


Figure 58.2-1 E-Ray block diagram

Customer CPU Interface (CIF)

Connects a customer specific Host CPU to the E-Ray IP-module via the Generic CPU Interface.

Generic CPU Interface (GIF)

The E-Ray IP-module is provided with an 8/16/32-bit Generic CPU Interface prepared for the connection to a wide range of customer-specific Host CPUs. Configuration registers, status registers, and interrupt registers are attached to the respective blocks and can be accessed via the Generic CPU Interface.

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The E-Ray Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- Transient Buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

MB91460 Series

Transient Buffer RAM (TBF A/B)

- Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Transient Buffer RAMs for intermediate message storage and to the physical layer via bus driver BD.

They perform the following functionality:

- Control and check of bit timing
- Reception / transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC
- Interfacing to bus driver

The FlexRay Channel Protocol Controllers have interfaces to:

- Physical Layer (bus driver)
- Transient Buffer RAM
- Message Handler
- Global Time Unit
- System Universal Control
- Frame and Symbol Processing
- Network Management
- Interrupt Control

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
 - rate correction
 - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation
- Monitor Mode

58.3. Programmer's Model

58.3.1 Register Map

The E-Ray module allocates an address space of 2 Kbytes (0xD000 to 0xD7FF). The registers are organized as 32-bit registers. 8/16-bit accesses are also supported. Host access to the Message RAM is done via the Input and Output Buffers. They buffer data to be transferred to and from the Message RAM under control of the Message Handler, avoiding conflicts between Host accesses and message reception / transmission. Addresses 0xD000 to 0xD00F are reserved for customer specific purposes. All functions related to these addresses are located in the Customer CPU Interface.

The assignment of the message buffers is done according to the scheme shown in [Table 58.3-1"Assignment of message buffers"](#) on P. 1256 below. The number N of available message buffers depends on the payload length of the configured message buffers. The maximum number of message buffers is 128. The maximum payload length supported is 254 bytes.

The message buffers are separated into three consecutive groups:

- Static Buffers - Transmit / receive buffers assigned to static segment
- Static + Dynamic Buffers - Transmit / receive buffers assigned to static or dynamic segment
- FIFO - Receive FIFO

The message buffer separation configuration can be changed only in DEFAULT_CONFIG or CONFIG state only by programming register MRC (see ["Message RAM Configuration \(MRC\)"](#) on P. 1311).

The first group starts with message buffer 0 and consists of static message buffers only. Message buffer 0 is dedicated to hold the startup / sync frame or the single slot frame, if the node transmits one, as configured by **SUCC1.TXST**, **SUCC1.TXSY**, and **SUCC1.TSM**. In addition, message buffer 1 may be used for sync frame transmission in case that sync frames or single-slot frames should have different payloads on the two channels. In this case bit **MRC.SPLM** has to be programmed to '1' and message buffers 0 and 1 have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only.

The second group consists of message buffers assigned to the static or to the dynamic segment. Message buffers belonging to this group may be reconfigured during run time from dynamic to static or vice versa depending on the state of **MRC.SEC[1:0]**.

The message buffers belonging to the third group are concatenated to a single receive FIFO.

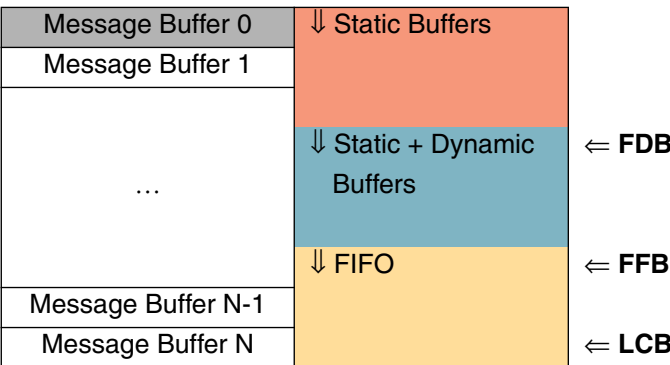


Table 58.3-1 Assignment of message buffers

MB91460 Series

Address	Symbol	Name	Page	Reset	Acc	Block
Customer Registers						
0xD000	CIF0	Version	P. 1262	04FF 5BFF	r	CIF
0xD004	CIF1	Customer Interface Register 1	P. 1262	0000 0000	r/w	
0xD008 - 0xD00C		reserved (2)		0000 0000	r0	
Special Registers						
0xD010 - 0xD018		reserved (3)		0000 0000	r	
0xD01C	LCK	Lock Register	P. 1264	0000 0000	r/w	GIF
Interrupt Registers						
0xD020	EIR	Error Interrupt Register	P. 1265	0000 0000	r/w	INT
0xD024	SIR	Status Interrupt Register	P. 1268	0000 0000	r/w	
0xD028	EILS	Error Interrupt Line Select	P. 1271	0000 0000	r/w	
0xD02C	SILS	Status Interrupt Line Select	P. 1272	0303 FFFF	r/w	
0xD030	EIES	Error Interrupt Enable Set	P. 1273	0000 0000	r/w	
0xD034	EIER	Error Interrupt Enable Reset	P. 1273	0000 0000	r/w	
0xD038	SIES	Status Interrupt Enable Set	P. 1274	0000 0000	r/w	
0xD03C	SIER	Status Interrupt Enable Reset	P. 1274	0000 0000	r/w	
0xD040	ILE	Interrupt Line Enable	P. 1275	0000 0000	r/w	
0xD044	T0C	Timer 0 Configuration	P. 1276	0000 0000	r/w	
0xD048	T1C	Timer 1 Configuration	P. 1277	0002 0000	r/w	
0xD04C	STPW1	Stop Watch Register 1	P. 1278	0000 0000	r/w	
0xD050	STPW2	Stop Watch Register 2	P. 1279	0000 0000	r/w	
0xD054 - 0xD07C		reserved (11)		0000 0000	r	
CC Control Registers						
0xD080	SUCC1	SUC Configuration Register 1	P. 1280	0C40 1080	r/w	SUC
0xD084	SUCC2	SUC Configuration Register 2	P. 1286	0100 0504	r/w	
0xD088	SUCC3	SUC Configuration Register 3	P. 1286	0000 0011	r/w	
0xD08C	NEMC	NEM Configuration Register	P. 1287	0000 0000	r/w	NEM
0xD090	PRTC1	PRT Configuration Register 1	P. 1287	084C 0633	r/w	PRT
0xD094	PRTC2	PRT Configuration Register 2	P. 1288	0F2D 0A0E	r/w	
0xD098	MHDC	MHD Configuration Register	P. 1288	0000 0000	r/w	MHD
0xD09C		reserved (1)		0000 0000	r	

Address	Symbol	Name	Page	Reset	Acc	Block
0xD0A0	GTUC1	GTU Configuration Register 1	P. 1290	0000 0280	r/w	GTU
0xD0A4	GTUC2	GTU Configuration Register 2	P. 1290	0002 000A	r/w	
0xD0A8	GTUC3	GTU Configuration Register 3	P. 1291	0202 0000	r/w	
0xD0AC	GTUC4	GTU Configuration Register 4	P. 1292	0008 0007	r/w	
0xD0B0	GTUC5	GTU Configuration Register 5	P. 1292	0E00 0000	r/w	
0xD0B4	GTUC6	GTU Configuration Register 6	P. 1293	0002 0000	r/w	
0xD0B8	GTUC7	GTU Configuration Register 7	P. 1294	0002 0004	r/w	
0xD0BC	GTUC8	GTU Configuration Register 8	P. 1294	0000 0002	r/w	
0xD0C0	GTUC9	GTU Configuration Register 9	P. 1295	0000 0101	r/w	
0xD0C4	GTUC10	GTU Configuration Register 10	P. 1295	0002 0005	r/w	
0xD0C8	GTUC11	GTU Configuration Register 11	P. 1296	0000 0000	r/w	
0xD0CC - 0xD0FC		reserved (13)		0000 0000	r	
CC Status Registers						
0xD100	CCSV	CC Status Vector	P. 1297	0010 4000	r	SUC
0xD104	CCEV	CC Error Vector	P. 1300	0000 0000	r	
0xD108 - 0xD10C		reserved (2)		0000 0000	r	
0xD110	SCV	Slot Counter Value	P. 1301	0000 0000	r	GTU
0xD114	MTCCV	Macrotick and Cycle Counter Value	P. 1301	0000 0000	r	
0xD118	RCV	Rate Correction Value	P. 1302	0000 0000	r	
0xD11C	OCV	Offset Correction Value	P. 1302	0000 0000	r	
0xD120	SFS	Sync Frame Status	P. 1303	0000 0000	r	
0xD124	SWNIT	Symbol Window and NIT Status	P. 1304	0000 0000	r	
0xD128	ACS	Aggregated Channel Status	P. 1306	0000 0000	r/w	
0xD12C		reserved (1)		0000 0000	r	
0xD130 - 0xD168	ESIDn	Even Sync ID [1...15]	P. 1308	0000 0000	r	
0xD16C		reserved (1)		0000 0000	r	
0xD170 - 0xD1A8	OSIDn	Odd Sync ID [1...15]	P. 1309	0000 0000	r	
0xD1AC		reserved (1)		0000 0000	r	
0xD1B0 - 0xD1B8	NMVn	Network Management Vector [1...3]	P. 1310	0000 0000	r	NEM
0xD1BC - 0xD2FC		reserved (81)		0000 0000	r	
Message Buffer Control Registers						
0xD300	MRC	Message RAM Configuration	P. 1311	0180 0000	r/w	MHD
0xD304	FRF	FIFO Rejection Filter	P. 1313	0180 0000	r/w	
0xD308	FRFM	FIFO Rejection Filter Mask	P. 1314	0000 0000	r/w	
0xD30C	FCL	FIFO Critical Level	P. 1314	0000 0080	r/w	

MB91460 Series

Address	Symbol	Name	Page	Reset	Acc	Block
Message Buffer Status Registers						
0xD310	MHDS	Message Handler Status	P. 1315	0000 0080	r/w	MHD
0xD314	LDTS	Last Dynamic Transmit Slot	P. 1316	0000 0000	r	
0xD318	FSR	FIFO Status Register	P. 1317	0000 0000	r	
0xD31C	MHDF	Message Handler Constraints Flags	P. 1318	0000 0000	r/w	
0xD320	TXRQ1	Transmission Request 1	P. 1320	0000 0000	r	
0xD324	TXRQ2	Transmission Request 2	P. 1320	0000 0000	r	
0xD328	TXRQ3	Transmission Request 3	P. 1320	0000 0000	r	
0xD32C	TXRQ4	Transmission Request 4	P. 1320	0000 0000	r	
0xD330	NDAT1	New Data 1	P. 1321	0000 0000	r	
0xD334	NDAT2	New Data 2	P. 1321	0000 0000	r	
0xD338	NDAT3	New Data 3	P. 1321	0000 0000	r	
0xD33C	NDAT4	New Data 4	P. 1321	0000 0000	r	
0xD340	MBSC1	Message Buffer Status Changed 1	P. 1322	0000 0000	r	
0xD344	MBSC2	Message Buffer Status Changed 2	P. 1322	0000 0000	r	
0xD348	MBSC3	Message Buffer Status Changed 3	P. 1322	0000 0000	r	
0xD34C	MBSC4	Message Buffer Status Changed 4	P. 1322	0000 0000	r	
0xD350 - 0xD3EC		reserved (40)		0000 0000	r	
Identification Registers						
0xD3F0	CREL	Core Release Register	P. 1323	[release info]	r	GIF
0xD3F4	ENDN	Endian Register	P. 1324	8765 4321	r	
0xD3F8 - 0xD3FC		reserved (2)		0000 0000	r	
Input Buffer						
0xD400 - 0xD4FC	WRDSn	Write Data Section [1...64]	P. 1324	0000 0000	r/w	IBF
0xD500	WRHS1	Write Header Section 1	P. 1325	0000 0000	r/w	
0xD504	WRHS2	Write Header Section 2	P. 1326	0000 0000	r/w	
0xD508	WRHS3	Write Header Section 3	P. 1327	0000 0000	r/w	
0xD50C		reserved (1)		0000 0000	r/w	
0xD510	IBCM	Input Buffer Command Mask	P. 1327	0000 0000	r/w	
0xD514	IBCR	Input Buffer Command Request	P. 1328	0000 0000	r/w	
0xD518 - 0xD5FC		reserved (58)		0000 0000	r	

Address	Symbol	Name	Page	Reset	Acc	Block
Output Buffer						
0xD600 - 0xD6FC	RDDSn	Read Data Section [1...64]	P. 1329	0000 0000	r	OBF
0xD700	RDHS1	Read Header Section 1	P. 1330	0000 0000	r	
0xD704	RDHS2	Read Header Section 2	P. 1331	0000 0000	r	
0xD708	RDHS3	Read Header Section 3	P. 1332	0000 0000	r	
0xD70C	MBS	Message Buffer Status	P. 1333	0000 0000	r	
0xD710	OBCM	Output Buffer Command Mask	P. 1336	0000 0000	r/w	
0xD714	OBCR	Output Buffer Command Request	P. 1337	0000 0000	r/w	
0xD718 - 0xD7FC		reserved (58)		0000 0000	r	

Table 58.3-2 E-Ray register map

58.3.2 Customer Registers

The address space from 0xD000 to 0xD00F is reserved for customer-specific registers. These registers, are located in the Customer CPU Interface block.

Specification of the CIF registers is as follows:

CIF0 = 0xD000, CIF1 = 0xD004, CIF2 = 0xD008 und CIF3 = 0xD00C

Customer Interface Logic

CIF0	31																													0	
	VERSION																														
	R																														
CIF1	31	30	29	28	27	26	25	24	bit																						
	7	6	5	4	3	2	1	0																							
	DREQO	DLVLO	DMODO	DENBO	DREQI	DLVLI	DMODI	DENBI																							
	0	0	0	0	0	0	0	0	Initial																						
	R/W(RM1)	R/W	R/W	R/W	R/W(RM1)	R/W	R/W	R/W	Attribute																						
CIF1	23	22	21	20	19	18	17	16	bit																						
	7	6	5	4	3	2	1	0																							
				MASK4	MASK3	MASK2	MASK1	MASK0																							
	0	0	0	0	0	0	0	0	Initial																						
	R0/W0	R0/W0	R0/W0	R/W	R/W(RM1)	R/W	R/W	R/W	Attribute																						
CIF1	15	14	13	12	11	10	9	8	bit																						
	7	6	5	4	3	2	1	0																							
	RTEST			SWAP	TREQ1	TENB1	TREQ0	TENB0																							
	0	0	0	0	0	0	0	0	Initial																						
	R/W	R0/W0	R0/W0	R/W	R/W(RM1)	R/W	R/W(RM1)	R/W	Attribute																						

- R Read only register
- R/W Read/write register
- R/W(RM1) Read/write register (outputs '1' on RMW)
- R0/W0 Read only register (always outputs '0', write '0' is recommended)

All other CIF registers (CIF2 and CIF3) not mentioned here are R0/W0.

parameter VERSION = 32'h04_FF_5B_FF ;
0x04: Fujitsu
0xFF: see Boot-ROM Device-ID
0x5B: FR:91(0x5B), FX:96(0x60)
0xFF: see E-Ray-ID

■ CIF register functions

DENBI: DMA request enable on IBF
DENBO: DMA request enable on OBF

DENBx = 0 : DMA request is disabled
DENBx = 1 : DMA request is enabled

DMODI: DMA request mode on IBF
DMODO: DMA request mode on OBF

DMODx = 0 : DMA request mode is **eray_ibusy/eray_obusy** level mode
DMODx = 1 : DMA request mode is **eray_ibusy/eray_obusy** edge mode

DLVLI: DMA level/edge selector on IBF
DLVLO: DMA level/edge selector on OBF

with DMODx = 0
DLVLx = 0 : DMA request level is non-inverted **eray_ibusy/eray_obusy**
DLVLx = 1 : DMA request level is inverted **eray_ibusy/eray_obusy**

with DMODx = 1
DLVLx = 0 : DMA request is negative edge of **eray_ibusy/eray_obusy**
DLVLx = 1 : DMA request is positive edge of **eray_ibusy/eray_obusy**

DREQI: DMA request flag on IBF
DREQO: DMA request flag on OBF

with DMODx = 0
DREQx = Read only, displays the **eray_ibusy/eray_obusy** level
- displays the modified **eray_ibusy/eray_obusy** level if changed with DLVLx
- On a read-modify-write bit-operation '1' is read

with DMODx = 1
DREQx = 0 : DMA request is inactive
DREQx = 1 : DMA request is active
Request is automatically cleared to '0' if a DMA transfer has started
- Possible to write '0' to clear the DMA request by the CPU
- On a read-modify-write bit-operation '1' is read

MASK0: DMA Channel 0 Interrupt Mask (for OBF configuration)
MASK1: DMA Channel 1 Interrupt Mask (for OBF configuration)
MASK2: DMA Channel 2 Interrupt Mask (for OBF configuration)
MASK3: DMA Channel 3 Interrupt Mask (for OBF configuration)
MASK4: DMA Channel 4 Interrupt Mask (for OBF configuration)

MASKx = 0 : DMA channel x interrupt is not masked
MASKx = 1 : DMA channel x interrupt is masked while **eray_obusy = 1**

TENB0: Timer 0 Interrupt Enable
TENB1: Timer 1 Interrupt Enable

TENBx = 0 : Direct **FlexRay Timer interrupt 0/FlexRay Timer interrupt 1** signal is used for interrupt generation

MB91460 Series

TENBx = 1 : Registered TREQx flag is used for interrupt generation

TREQ0: Timer 0 Interrupt Request

TREQ1: Timer 1 Interrupt Request

with TENBx = 0

TREQx = Read only, displays the **FlexRay Timer interrupt 0/FlexRay Timer interrupt 1** level

- On a read-modify-write bit-operation '1' is read

with TENBx = 1

TREQx = 0 : Timer interrupt request is inactive

TREQx = 1 : Timer interrupt request is active

- Possible to write '0' to clear the interrupt request by the CPU

- On a read-modify-write bit-operation '1' is read

SWAP: IBF/OBF data swap enable

SWAP = 0 : Read and write data on IBF/OBF is not swapped

SWAP = 1 : Read and write data on IBF/OBF is swapped

SWAP = 0	SWAP = 1
MD[7: 0] = DW(n), byte(n-1)	MD[7: 0] = DW(n+1), byte(n+2)
MD[15: 8] = DW(n), byte(n)	MD[15: 8] = DW(n+1), byte(n+1)
MD[23:16] = DW(n+1), byte(n+1)	MD[23:16] = DW(n), byte(n)
MD[31:24] = DW(n+1), byte(n+2)	MD[31:24] = DW(n), byte(n-1)

RTEST: RAM Test address range enable (ONLY FOR TESTMODE)

RTEST = 0 : Normal operation address mapping

RTEST = 1 : RAM Test operation address mapping (use when TMC[1:0]=01)

58.3.3 Special Registers

■ Lock Register (LCK)

The Lock Register is write-only. Reading the register will return 0x0000 0000.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCK	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD01C	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD01C	W								CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CLK[7:0] Configuration Lock Key

To leave CONFIG state by writing **SUCC1.CMD[3:0]** (commands READY, MONITOR_MODE, ATM), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the SUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: **LCK.CLK[7:0]** = "1100 1110" (0xCE)

Second write: **LCK.CLK[7:0]** = "0011 0001" (0x31)

Third write: **SUCC1.CMD[3:0]**

MB91460 Series

58.3.4 Interrupt Registers

■ Error Interrupt Register (EIR)

The flags are set when the CC detects one of the listed error conditions. The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hard reset will also clear the register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIR	R	0	0	0	0	TABB	LTVB	EDB	0	0	0	0	0	TABA	LTVA	EDA
	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

PEMC POC Error Mode Changed

This flag is set whenever the error mode signalled by **CCEV.ERRM[1:0]** has changed.

1 = Error mode has changed

0 = Error mode has not changed

CNA Command Not Accepted

The flag signals that the write access to the CHI command vector **SUCC1.CMD[3:0]** was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (**CCL** = '1').

1 = CHI command not accepted

0 = CHI command accepted

SFBM Sync Frames Below Minimum

This flag signals that the number of sync frames received during the last communication cycle was below the limit required by the FlexRay protocol. May be set during startup and therefore should be cleared by the Host after the CC entered **NORMAL_ACTIVE** state.

1 = Less than the required minimum of sync frames received

0 = Sync node: 1 or more sync frames received, non-sync node: 2 or more sync frames received

SFO Sync Frame Overflow

Set when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by **GTUC2.SNM[3:0]**.

1 = More sync frames received than configured by **GTUC2.SNM[3:0]**

0 = Number of received sync frames ≤ **GTUC2.SNM[3:0]**

CCF Clock Correction Failure

This flag is set at the end of the cycle whenever one of the following errors occurred:

- Missing offset and / or rate correction
- Clock correction limit reached

The clock correction status is monitored in registers CCEV and SFS. A failure may occur during startup, therefore bit **CCF** should be cleared by the Host after the CC entered NORMAL_ACTIVE state.

- 1 = Clock correction failed
- 0 = No clock correction error

CCL CHI Command Locked

The flag signals that the write access to the CHI command vector **SUCC1.CMD[3:0]** was not successful because the execution of the previous CHI command has not yet completed. In this case bit **CNA** is also set to '1'.

- 1 = CHI command not accepted
- 0 = CHI command accepted

PERR Parity Error

The flag signals a parity error to the Host. It is set whenever one of the flags **MHDS.PIBF**, **MHDS.POBF**, **MHDS.PMR**, **MHDS.PTBF1**, **MHDS.PTBF2** changes from '0' to '1'.

- 1 = Parity error detected
- 0 = No parity error detected

RFO Receive FIFO Overrun

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FSR.

- 1 = A receive FIFO overrun has been detected
- 0 = No receive FIFO overrun detected

EFA Empty FIFO Access

This flag is set by the CC when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty.

- 1 = Host access to empty FIFO occurred
- 0 = No Host access to empty FIFO occurred

IIBA Illegal Input Buffer Access

This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies:

- 1) The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the
 - Header section of message buffer 0, 1 if configured for transmission in key slot
 - Header section of static message buffers with buffer number < **MRC.FDB[7:0]** while **MRC.SEC[1:0]** = "01"
 - Header section of any static or dynamic message buffer while **MRC.SEC[1:0]** = "1x"
 - Header and / or data section of any message buffer belonging to the receive FIFO
 - 2) The Host writes to any register of the Input Buffer while **IBCR.IBSYH** is set to '1'.
- 1 = Illegal Host access to Input Buffer occurred

MB91460 Series

0 = No illegal Host access to Input Buffer occurred

IOBA Illegal Output buffer Access

This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while **OBCR.OBSYS** is set to '1'.

1 = Illegal Host access to Output Buffer occurred

0 = No illegal Host access to Output Buffer occurred

MHF Message Handler Constraints Flag

The flag signals a Message Handler constraints violation condition. It is set whenever one of the flags **MHDF.SNUA**, **MHDF.SNUB**, **MHDF.FNFA**, **MHDF.FNFB**, **MHDF.TBFA**, **MHDF.TBFB**, **MHDF.WAHP** changes from '0' to '1'.

1 = Message Handler failure detected

0 = No Message Handler failure detected

Channel-specific error flags:

EDA Error Detected on Channel A

This bit is set whenever one of the flags **ACS.SEDA**, **ACS.CEDA**, **ACS.CIA**, **ACS.SBVA** changes from '0' to '1'.

1 = Error detected on channel A

0 = No error detected on channel A

LTVA Latest Transmit Violation Channel A

The flag signals a latest transmit violation on channel A to the Host.

1 = Latest transmit violation detected on channel A

0 = No latest transmit violation detected on channel A

TABA Transmission Across Boundary Channel A

The flag signals to the Host that a transmission across a slot boundary occurred for channel A.

1 = Transmission across slot boundary detected on channel A

0 = No transmission across slot boundary detected on channel A

EDB Error Detected on Channel B

This bit is set whenever one of the flags **ACS.SEDB**, **ACS.CEDB**, **ACS.CIB**, **ACS.SBVB** changes from '0' to '1'.

1 = Error detected on channel B

0 = No error detected on channel B

LTVB Latest Transmit Violation Channel B

The flag signals a latest transmit violation on channel B to the Host.

1 = Latest transmit violation detected on channel B

0 = No latest transmit violation detected on channel B

TABB Transmission Across Boundary Channel B

The flag signals to the Host that a transmission across a slot boundary occurred for channel B.

1 = Transmission across slot boundary detected on channel B

0 = No transmission across slot boundary detected on channel B

■ Status Interrupt Register (SIR)

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hard reset will also clear the register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIR	R	0	0	0	0	0	0	MTSB	WUPB	0	0	0	0	0	0	0
0xD024	W														MTSA	WUPA
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS
	W															WST
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WST Wakeup Status

This flag is set when **CCSV.WSV[2:0]** changes to a value other than UNDEFINED.

1 = Wakeup status changed

0 = Wakeup status unchanged

CAS Collision Avoidance Symbol

This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received.

1 = Bit pattern matching the CAS symbol received

0 = No bit pattern matching the CAS symbol received

CYCS Cycle Start Interrupt

This flag is set by the CC when a communication cycle starts.

1 = Communication cycle started

0 = No communication cycle started

TXI Transmit Interrupt

This flag is set by the CC at the end of frame transmission if bit **MBI** in the respective message buffer is set to '1' (see [Table 58.4-12 Header section of a message buffer in the Message RAM](#)).

1 = At least one frame was transmitted from a transmit buffer with **MBI** = '1'

0 = No frame transmitted from a transmit buffer with **MBI** = '1'

RXI Receive Interrupt

This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see [“New Data 1/2/3/4 \(NDAT1/2/3/4\)” on page 1321](#)), and if bit **MBI** of that message buffer is set to '1' (see [Table 58.4-12 Header section of a message buffer in the Message RAM](#)).

1 = At least one ND flag of a receive buffer with **MBI** = '1' has been set to '1'

0 = No ND flag of a receive buffer with **MBI** = '1' has been set to '1'

RFNE Receive FIFO Not Empty

This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register FSR.

1 = Receive FIFO is not empty

0 = Receive FIFO is empty

MB91460 Series**RFCL** Receive FIFO Critical Level

This flag is set when the receive FIFO fill level **FSR.RFFL[7:0]** is equal or greater than the critical level as configured by **FCL.CL[7:0]**.

1 = Receive FIFO critical level reached

0 = Receive FIFO below critical level

NMVC Network Management Vector Changed

This interrupt flag signals a change in the Network Management Vector visible to the Host.

1 = Network management vector changed

0 = No change in the network management vector

TI0 Timer Interrupt 0

This flag is set whenever timer 0 matches the conditions configured in register T0C. A Timer Interrupt 0 is also signalled on **FlexRay Timer interrupt 0**.

1 = Timer Interrupt 0 occurred

0 = No Timer Interrupt 0

TI1 Timer Interrupt 1

This flag is set whenever timer 1 matches the conditions configured in register T1C. A Timer Interrupt 1 is also signalled on **FlexRay Timer interrupt 1**.

1 = Timer Interrupt 1 occurred

0 = No Timer Interrupt 1

TIBC Transfer Input Buffer Completed

This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and **IBCR.IBSYS** has been reset by the Message Handler.

1 = Transfer between Input Buffer and Message RAM completed

0 = No transfer completed

TOBC Transfer Output Buffer Completed

This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and **OBCR.OBSYS** has been reset by the Message Handler.

1 = Transfer between Message RAM and Output Buffer completed

0 = No transfer completed

SWE Stop Watch Event

This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (see section “[Stop Watch Register 1 \(STPW1\)](#)” on page 1278).

1 = Stop Watch Event occurred

0 = No Stop Watch Event

SUCS Startup Completed Successfully

This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

1 = Startup completed successfully

0 = No startup completed successfully

MBSI Message Buffer Status Interrupt

This flag is set by the CC when the message buffer status **MBS** has changed and if bit **MBI** of that message buffer is set (see [Table 58.4-12 Header section of a message buffer in the Message RAM](#)).

- 1 = Message buffer status of at least one message buffer with **MBI** = '1' has changed
- 0 = No message buffer status change of message buffer with **MBI** = '1'

SDS Start of Dynamic Segment

This flag is set by the CC when the dynamic segment starts.

- 1 = Dynamic segment started
- 0 = Dynamic segment not yet started

Channel-specific status flags:

WUPA Wakeup Pattern Channel A

This flag is set by the CC when a wakeup pattern was received on channel A. Only set when the CC is in WAKEUP, READY, or STARTUP state, or when in Monitor mode.

- 1 = Wakeup pattern received on channel A
- 0 = No wakeup pattern received on channel A

MTSA MTS Received on Channel A (vSS!ValidMTSA)

Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

- 1 = MTS symbol received on channel A
- 0 = No MTS symbol received on channel A

WUPB Wakeup Pattern Channel B

This flag is set by the CC when a wakeup pattern was received on channel B. Only set when the CC is in WAKEUP, READY, or STARTUP state, or when in Monitor mode.

- 1 = Wakeup pattern received on channel B
- 0 = No wakeup pattern received on channel B

MTSB MTS Received on Channel B (vSS!ValidMTSB)

Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

- 1 = MTS symbol received on channel B
- 0 = No MTS symbol received on channel B

MB91460 Series**■ Error Interrupt Line Select (EILS)**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
EILS	R	0	0	0	0	0	TABBL	LTVBL	EDBL	0	0	0	0	0	TABAL	LTVAL	EDAL
0xD028	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

The Error Interrupt Line Select register assigns an interrupt generated by a specific error interrupt flag from register EIR to one of the two module interrupt lines:

- 1 = Interrupt assigned to the FlexRay channel 1 interrupt
- 0 = Interrupt assigned to the FlexRay channel 0 interrupt

PEMCL	POC Error Mode Changed Interrupt Line
CNAL	Command Not Accepted Interrupt Line
SFBML	Sync Frames Below Minimum Interrupt Line
SFOL	Sync Frame Overflow Interrupt Line
CCFL	Clock Correction Failure Interrupt Line
CCLL	CHI Command Locked Interrupt Line
PERRL	Parity Error Interrupt Line
RFOL	Receive FIFO Overrun Interrupt Line
EFAL	Empty FIFO Access Interrupt Line
IIBAL	Illegal Input Buffer Access Interrupt Line
IOBAL	Illegal Output Buffer Access Interrupt Line
MHFL	Message Handler Constraints Flag Interrupt Line
EDAL	Error Detected on Channel A Interrupt Line
LTVAL	Latest Transmit Violation Channel A Interrupt Line
TABAL	Transmission Across Boundary Channel A Interrupt Line
EDBL	Error Detected on Channel B Interrupt Line
LTVBL	Latest Transmit Violation Channel B Interrupt Line
TABBL	Transmission Across Boundary Channel B Interrupt Line

■ Status Interrupt Line Select (SILS)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILS	R	0	0	0	0	0	0	MTSBL	WUPBL	0	0	0	0	0	0	0
0xD02C	W														MTSAL	WUPAL
Reset		0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL
	W															WSTL
Reset		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The Status Interrupt Line Select register assign an interrupt generated by a specific status interrupt flag from register SIR to one of the two module interrupt lines:

- 1 = Interrupt assigned to the FlexRay channel 1 interrupt
- 0 = Interrupt assigned to the FlexRay channel 0 interrupt

WSTL	Wakeup Status Interrupt Line
CASL	Collision Avoidance Symbol Interrupt Line
CYCSL	Cycle Start Interrupt Line
TXIL	Transmit Interrupt Line
RXIL	Receive Interrupt Line
RFNEL	Receive FIFO Not Empty Interrupt Line
RFCLL	Receive FIFO Critical Level Interrupt Line
NMVCL	Network Management Vector Changed Interrupt Line
TI0L	Timer Interrupt 0 Line
TI1L	Timer Interrupt 1 Line
TIBCL	Transfer Input Buffer Completed Interrupt Line
TOBCL	Transfer Output Buffer Completed Interrupt Line
SWEL	Stop Watch Event Interrupt Line
SUCSL	Startup Completed Successfully Interrupt Line
MBSIL	Message Buffer Status Interrupt Line
SDSL	Start of Dynamic Segment Interrupt Line
WUPAL	Wakeup Pattern Channel A Interrupt Line
MTSAL	Media Access Test Symbol Channel A Interrupt Line
WUPBL	Wakeup Pattern Channel B Interrupt Line
MTSBL	Media Access Test Symbol Channel B Interrupt Line

MB91460 Series**■ Error Interrupt Enable Set / Reset (EIES, EIER)**

The settings in the Error Interrupt Enable register determine which status changes in the Error Interrupt Register will result in an interrupt.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIES,R	R	0	0	0	0	0			0	0	0	0	0			
EIES:0xD030 EIER:0xD034	W						TABBE	LTVBE	EDBE					TABAE	LTVAE	EDAE
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0												
W					MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The enable bits are set by writing to address 0xD030 and reset by writing to address 0xD034. Writing a '1' sets / resets the specific enable bit, writing a '0' has no effect. Reading from both addresses will result in the same value.

1 = Interrupt enabled

0 = Interrupt disabled

PEMCE	POC Error Mode Changed Interrupt Enable
CNAE	Command Not Accepted Interrupt Enable
SFBME	Sync Frames Below Minimum Interrupt Enable
SFOE	Sync Frame Overflow Interrupt Enable
CCFE	Clock Correction Failure Interrupt Enable
CCLE	CHI Command Locked Interrupt Enable
PERRE	Parity Error Interrupt Enable
RFOE	Receive FIFO Overrun Interrupt Enable
EFAE	Empty FIFO Access Interrupt Enable
IIBAE	Illegal Input Buffer Access Interrupt Enable
IOBAE	Illegal Output Buffer Access Interrupt Enable
MHFE	Message Handler Constraints Flag Interrupt Enable
EDAE	Error Detected on Channel A Interrupt Enable
LTVAE	Latest Transmit Violation Channel A Interrupt Enable
TABAE	Transmission Across Boundary Channel A Interrupt Enable
EDBE	Error Detected on Channel B Interrupt Enable
LTVBE	Latest Transmit Violation Channel B Interrupt Enable
TABBE	Transmission Across Boundary Channel B Interrupt Enable

■ Status Interrupt Enable Set / Reset (SIES, SIER)

The settings in the Status Interrupt Enable register determine which status changes in the Status Interrupt Register will result in an interrupt.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIES,R	R	0	0	0	0	0	0		0	0	0	0	0	0		
SIES:0xD038 SIER:0xD03C	W						MTSBE	WUPBE							MTSAE	WUPAE
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE
	W															WSTE
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The enable bits are set by writing to address 0xD038 and reset by writing to address 0xD03C. Writing a '1' sets / resets the specific enable bit, writing a '0' has no effect. Reading from both addresses will result in the same value.

1 = Interrupt enabled

0 = Interrupt disabled

WSTE	Wakeup Status Interrupt Enable
CASE	Collision Avoidance Symbol Interrupt Enable
CYCSE	Cycle Start Interrupt Enable
TXIE	Transmit Interrupt Enable
RXIE	Receive Interrupt Enable
RFNEE	Receive FIFO Not Empty Interrupt Enable
RFCLE	Receive FIFO Critical Level Interrupt Enable
NMVCE	Network Management Vector Changed Interrupt Enable
TI0E	Timer Interrupt 0 Enable
TI1E	Timer Interrupt 1 Enable
TIBCE	Transfer Input Buffer Completed Interrupt Enable
TOBCE	Transfer Output Buffer Completed Interrupt Enable
SWEE	Stop Watch Event Interrupt Enable
SUCSE	Startup Completed Successfully Interrupt Enable
MBSIE	Message Buffer Status Interrupt Enable
SDSE	Start of Dynamic Segment Interrupt Enable
WUPAE	Wakeup Pattern Channel A Interrupt Enable
MTSAE	MTS Received on Channel A Interrupt Enable
WUPBE	Wakeup Pattern Channel B Interrupt Enable
MTSBE	MTS Received on Channel B Interrupt Enable

■ Interrupt Line Enable (ILE)

Each of the two interrupt lines to the Host (**FlexRay Channel 0**, **FlexRay Channel 1**) can be enabled / disabled separately by programming bit **EINT0** and **EINT1**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ILE	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD040	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0		
	W														EINT1	EINT0
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EINT0 Enable Interrupt Line 0

- 1 = Interrupt line **FlexRay Channel 0** enabled
- 0 = Interrupt line **FlexRay Channel 0** disabled

EINT1 Enable Interrupt Line 1

- 1 = Interrupt line **FlexRay Channel 1** enabled
- 0 = Interrupt line **FlexRay Channel 1** disabled

■ Timer 0 Configuration (T0C)

Absolute timer. Specifies in terms of cycle count and macrotick the point in time when the timer 0 interrupt occurs. When the timer 0 interrupt is asserted, output signal **FlexRay Timer interrupt 0** is set to '1' for the duration of one macrotick and **SIR.TI0** is set to '1'.

Timer 0 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit **T0RC** to '0'.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T0C	R	0	0	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO	T0MO
0xD044	W			13	12	11	10	9	8	7	6	5	4	3	2	1
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	T0CC6	T0CC5	T0CC4	T0CC3	T0CC2	T0CC1	T0CC0	0	0	0	0	0	0	
	W														T0MS	T0RC
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

T0RC Timer 0 Run Control

- 1 = Timer 0 running
- 0 = Timer 0 halted

T0MS Timer 0 Mode Select

- 1 = Continuous mode
- 0 = Single-shot mode

T0CC[6:0] Timer 0 Cycle Code

The 7-bit timer 0 cycle code determines the cycle set used for generation of the timer 0 interrupt. For details about the configuration of the cycle code see Section [“Cycle Counter Filtering”](#) on page 1364.

T0MO[13:0] Timer 0 Macrotick Offset

Configures the macrotick offset from the beginning of the cycle where the interrupt is to occur. The Timer 0 Interrupt occurs at this offset for each cycle of the cycle set.

Note: The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0. In case the CC leaves NORMAL_ACTIVE or NORMAL_PASSIVE state, or if timer 0 is halted by Host command, output signal **FlexRay Timer interrupt 0** is reset to '0' immediately.

MB91460 Series

■ Timer 1 Configuration (T1C)

Relative timer. After the specified number of macroticks has expired, the timer 1 interrupt is asserted, output signal **FlexRay Timer interrupt 1** is set to '1' for the duration of one macrotick and **SIR.TI1** is set to '1'.

Timer 1 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit **T1RC** to '0'.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T1C	R	0	0	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC	T1MC
0xD048	W			13	12	11	10	9	8	7	6	5	4	3	2	1
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0		
	W														T1MS	T1RC
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

T1RC Timer 1 Run Control

1 = Timer 1 running

0 = Timer 1 halted

T1MS Timer 1 Mode Select

1 = Continuous mode

0 = Single-shot mode

T1MC[13:0] Timer 1 Macrotick Count

When the configured macrotick count is reached the timer 1 interrupt is generated.

Valid values are: 2 to 16383 MT in continuous mode

1 to 16383 MT in single-shot mode

Note: In case the CC leaves NORMAL_ACTIVE or NORMAL_PASSIVE state, or if timer 1 is halted by Host command, output signal **FlexRay Timer interrupt 1** is reset to '0' immediately.

■ Stop Watch Register 1 (STPW1)

The stop watch is activated by a rising or falling edge on pin **eray_stpwt**, by an interrupt 0,1 event (rising edge on pin **FlexRay Channel interrupt 0** or **FlexRay Channel interrupt 1**) or by the Host by writing bit **SSWT** to '1'. With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick values are captured in register STPW1 while the slot counter values for channel A and B are captured in register STPW2.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
STPW1	R	0	0	SMTV13	SMTV12	SMTV11	SMTV10	SMTV9	SMTV8	SMTV7	SMTV6	SMTV5	SMTV4	SMTV3	SMTV2	SMTV1	SMTV0
	W																
0xD04C																	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	SCCV5	SCCV4	SCCV3	SCCV2	SCCV1	SCCV0	0	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESWT Enable Stop Watch Trigger

If enabled an edge on input **eray_stpwt** or an interrupt 0,1 event (rising edge on pin **FlexRay Channel interrupt 0** or **FlexRay Channel interrupt 1**) activates the stop watch. In single-shot mode this bit is reset to '0' after the actual cycle counter and macrotick value are stored in the Stop Watch register.

1 = Stop watch trigger enabled

0 = Stop watch trigger disabled

SWMS Stop Watch Mode Select

1 = Continuous mode

0 = Single-shot mode

EDGE Stop Watch Trigger Edge Select

1 = Rising edge

0 = Falling edge

SSWT Software Stop Watch Trigger

When the Host writes this bit to '1' the stop watch is activated. After the actual cycle counter and macrotick value are stored in the Stop Watch register this bit is reset to '0'. The bit is only writeable while **ESWT** = '0'.

1 = Stop watch activated by software trigger

0 = Software trigger reset

EETP Enable External Trigger Pin

Enables stop watch trigger event via pin **eray_stpwt** if **ESWT** = '1'.

1 = Edge on pin **eray_stpwt** triggers stop watch

0 = Stop watch trigger via pin **eray_stpwt** disabled

EINT0 Enable Interrupt 0 Trigger

Enables stop watch trigger by interrupt 0 event if **ESWT** = '1'.

1 = Interrupt 0 event triggers stop watch

0 = Stop watch trigger by interrupt 0 disabled

MB91460 Series**EINT1** Enable Interrupt 1 Trigger

Enables stop watch trigger by interrupt 1 event if **ESWT** = '1'.

1 = Interrupt 1 event triggers stop watch

0 = Stop watch trigger by interrupt 1 disabled

SCCV[5:0] Stop Watch Captured Cycle Counter Value

State of the cycle counter when the stop watch event occurred. Valid values are 0 to 63.

SMTV[13:0] Stop Watch Captured Macrotock Value

State of the macrotock counter when the stop watch event occurred. Valid values are 0 to 15999.

Note: Bits **ESWT** and **SSWT** cannot be set to '1' simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

■ Stop Watch Register 2 (STPW2)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STPW2 R	0	0	0	0	0	SSCVB 10	SSCVB 9	SSCVB 8	SSCVB 7	SSCVB 6	SSCVB 5	SSCVB 4	SSCVB 3	SSCVB 2	SSCVB 1	SSCVB 0
0xD050 W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SSCVA 10	SSCVA 9	SSCVA 8	SSCVA 7	SSCVA 6	SSCVA 5	SSCVA 4	SSCVA 3	SSCVA 2	SSCVA 1	SSCVA 0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSCVA[10:0] Stop Watch Captured Slot Counter Value Channel A

State of the slot counter for channel A when the stop watch event occurred. Valid values are 0 to 2047.

SSCVB[10:0] Stop Watch Captured Slot Counter Value Channel B

State of the slot counter for channel B when the stop watch event occurred. Valid values are 0 to 2047.

58.3.5 CC Control Registers

This section describes the registers provided by the CC to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from hard reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to proceed as described in Section ["Lock Register \(LCK\)" on P. 1264](#).

All bits marked with an asterisk * can be updated in DEFAULT_CONFIG or CONFIG state only!

■ SUC Configuration Register 1 (SUCC1)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC1	R	0	0	0	0	CCHB*	CCHA*	MTSB*	MTSA*	HCSE*	TSM*	WUCS*	PTA4*	PTA3*	PTA2*	PTA1*	PTA0*
	W																
0xD080																	
Reset		0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	CSA4*	CSA3*	CSA2*	CSA1*	CSA0*	0	TXSY*	TXST*	PBSY	0	0	0	CMD3	CMD2	CMD1	CMD0
	W																
Reset		0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

CMD[3:0] CHI Command Vector

The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector **CMD[3:0]** will be reset to "0000" = command_not_accepted, and flag **EIR.CNA** will be set to '1'. In case the previous CHI command has not yet completed, **EIR.CCL** is set to '1' together with **EIR.CNA**; the CHI command needs to be repeated. Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will **EIR.CNA** be set.

0000 =command_not_accepted
0001 =CONFIG
0010 =READY
0011 =WAKEUP
0100 =RUN
0101 =ALL_SLOTS
0110 =HALT
0111 =FREEZE
1000 =SEND_MTS
1001 =ALLOW_COLDSTART
1010 =RESET_STATUS_INDICATORS
1011 =MONITOR_MODE
1100 =CLEAR_RAMs
1101 =reserved
1110 =reserved
1111 =reserved

MB91460 Series

Reading **CMD[3:0]** shows whether the last CHI command was accepted. The actual POC state is monitored by **CCSV.POCS[5:0]**. The reserved CHI commands belong to the hardware test functions.

In general the Host must check **SUCC1.PBSY** before writing a new CHI command.

command_not_accepted

CMD[3:0] is reset to "0000" due to one of the following conditions:

- Illegal command applied by the Host
- Host applied command to leave CONFIG state without preceding config lock key
- Host applied new command while execution of the previous Host command has not completed
- Host writes **command_not_accepted**

When **CMD[3:0]** is reset to "0000", **EIR.CNA** is set, and - if enabled - an interrupt is generated. Commands which are not accepted are not executed.

CONFIG

Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, READY, or in MONITOR_MODE. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

READY

Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

WAKEUP

Go to POC state WAKEUP when called in POC state READY. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

RUN

Go to POC state STARTUP when called in POC state READY. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

ALL_SLOTS

Leave SINGLE slot mode after successful startup / integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

HALT

Set halt request **CCSV.HRQ** and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, **CMD[3:0]** will be reset to "0000" = command_not_accepted.

FREEZE

Set the freeze status indicator **CCSV.FSI** and go to POC state HALT immediately. Can be called from any state.

SEND_MTS

Send single MTS symbol during the next following symbol window on the channel configured by **MTSA**, **MTSB**, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode

(**CCSV.SLM**[1:0] = "11"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, **CMD**[3:0] will be reset to "0000" = command_not_accepted.

ALLOW_COLDSTART

The command resets **CCSV.CSI** to enable the node to become leading coldstarter. When called in states **DEFAULT_CONFIG**, **CONFIG**, **HALT**, or **MONITOR_MODE**, **CMD**[3:0] will be reset to "0000" = command_not_accepted. To become leading coldstarter it is also required that both **TXST** and **TXSY** are set.

RESET_STATUS_INDICATORS

Resets status flags **CCSV.CSNI**, **CCSV.CSAI**, and **CCSV.WSV**[2:0] to their default values. May be called in POC states **READY** and **STARTUP**. When called in any other state, **CMD**[3:0] will be reset to "0000" = command_not_accepted.

MONITOR_MODE

Enter **MONITOR_MODE** when called in POC state **CONFIG**. In this mode the CC is able to receive FlexRay frames and wakeup pattern. It is also able to detect coding errors. The temporal integrity of received frames is not checked. This mode can be used for debugging purposes, e.g. in case that the startup of a FlexRay network fails. When called in any other state, **CMD**[3:0] will be reset to "0000" = command_not_accepted. For details see "[MONITOR_MODE](#)" on page 1350

CLEAR_RAMs

Sets **MHDS.CRAM** when called in **DEFAULT_CONFIG** or **CONFIG** state. When called in any other state, **CMD**[3:0] will be reset to "0000" = command_not_accepted. **MHDS.CRAM** is also set when the CC leaves hard reset. By setting **MHDS.CRAM** all internal RAM blocks are initialized to zero. During the initialization of the RAMs, **PBSY** will show POC busy. Access to the configuration and status registers is possible during execution of CHI command **CLEAR_RAMs**.

The initialization of the E-Ray internal RAM blocks requires 2048 **CLKB** cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after hard reset or after assertion of CHI command **CLEAR_RAMs**. Before asserting CHI command **CLEAR_RAMs** the Host should make sure that no transfer between Message RAM and IBF / OBF or the Transient Buffer RAMs is ongoing. This command also resets the Message Buffer Status registers **MHDS**, **LDTs**, **FSR**, **MHDF**, **TXRQ**1/2/3/4, **NDAT**1/2/3/4, and **MBSC**1/2/3/4.

Note: All accepted commands with exception of **CLEAR_RAMs** and **SEND_MTS** will cause a change of the POC state in the **ERAY_SCLK** domain after at most 8 cycles of the slower of the two clocks **CLKB** and **ERAY_SCLK**, counted from the falling edge of the CHI input signal **eray_select**, assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register **CCSV** will show data that is additionally delayed by synchronization from **ERAY_SCLK** to **CLKB** domain and by the Host-specific CPU interface. The maximum additional delay is 12 cycles of the slower of the two clocks **CLKB** and **ERAY_SCLK**.

PBSY POC Busy

Signals that the POC is busy and cannot accept a command from the Host. **CMD**[3:0] is locked against write accesses. Set to '1' after hard reset during initialization of internal RAM blocks.

1 = POC is busy, **CMD**[3:0] locked

0 = POC not busy, **CMD**[3:0] writeable

TXST Transmit Startup Frame in Key Slot ([pKeySlotUsedForStartup](#))

MB91460 Series

Defines whether the key slot is used to transmit startup frames. The bit can be modified in DEFAULT_CONFIG or CONFIG state only.

1 = Key slot used to transmit startup frame, node is leading or following coldstarter

0 = No startup frame transmission in key slot, node is non-coldstarter

TXSY Transmit Sync Frame in Key Slot ([pKeySlotUsedForSync](#))

Defines whether the key slot is used to transmit sync frames. The bit can be modified in DEFAULT_CONFIG or CONFIG state only.

1 = Key slot used to transmit sync frame, node is sync node

0 = No sync frame transmission in key slot, node is neither sync nor coldstart node

Note: The protocol requires that both bits **TXST** and **TXSY** are set for coldstart nodes.

CSA[4:0] Cold Start Attempts ([gColdStartAttempts](#))

Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node. It can be modified in DEFAULT_CONFIG or CONFIG state only. Must be identical in all nodes of a cluster. Valid values are 2 to 31.

PTA[4:0] Passive to Active ([pAllowPassiveToActive](#))

Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If set to "00000" the CC is **not** allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. It can be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 31 even / odd cycle pairs.

WUCS Wakeup Channel Select ([pWakeupChannel](#))

With this bit the Host selects the channel on which the CC sends the Wakeup pattern. The CC ignores any attempt to change the status of this bit when not in DEFAULT_CONFIG or CONFIG state.

1 = Send wakeup pattern on channel B

0 = Send wakeup pattern on channel A

TSM Transmission Slot Mode ([pSingleSlotEnabled](#))

Selects the initial transmission slot mode. In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit **MRC.SPLM**. In case **TSM** = '1', message buffer 0 respectively message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots. **TSM** is a configuration bit which can only be set / reset by the Host. The bit can be written in DEFAULT_CONFIG or CONFIG state only. The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing **CMD[3:0]** = "0101" in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by **CCSV.SLM[1:0]**.

1 = SINGLE Slot Mode (default after hard reset)

0 = ALL Slot Mode

HCSE Halt due to Clock Sync Error ([pAllowHaltDueToClock](#))

Controls the transition to HALT state due to a clock synchronization error. The bit can be modified in DEFAULT_CONFIG or CONFIG state only.

1 = CC will enter HALT state

0 = CC will enter / remain in NORMAL_PASSIVE

MTSA Select Channel A for MTS Transmission

The bit selects channel A for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state.

1 = Channel A selected for MTS transmission

0 = Channel A disabled for MTS transmission

MTSB Select Channel B for MTS Transmission

The bit selects channel B for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state.

1 = Channel B selected for MTS transmission

0 = Channel B disabled for MTS transmission

Note: **MTSA,B** may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to SUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in "[Lock Register \(LCK\)](#)" on page 1264. This may be combined with CHI command SEND_MTS. If both bits **MTSA** and **MTSB** are set to '1' an MTS symbol will be transmitted on both channels when requested by writing **CMD[3:0]** = "1000".

CCHA Connected to Channel A ([pChannels](#))

Configures whether the node is connected to channel A.

1 = Node connected to channel A (default after hard reset)

0 = Not connected to channel A

CCHB Connected to Channel B ([pChannels](#))

Configures whether the node is connected to channel B.

1 = Node connected to channel B (default after hard reset)

0 = Not connected to channel B

MB91460 Series

Table below references the CHI commands from the FlexRay Protocol Specification v2.1 (section 2.2.1.1, Table 2-2) to the E-Ray CHI command vector CMD[3:0].

CHI command	Where processed (POC States)	CHI Command Vector CMD[3:0]
ALL_SLOTS	POC:normal active, POC:normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC:default config, POC:config, POC:halt	ALLOW_COLDSTART
CONFIG	POC:default config, POC:ready	CONFIG
CONFIG_COMPLETE	POC:config	Unlock sequence & READY
DEFAULT_CONFIG	POC:halt	CONFIG
FREEZE	All	FREEZE
HALT	POC:normal active, POC:normal passive	HALT
READY	All except POC:default config, POC:config, POC:ready, POC:halt	READY
RUN	POC:ready	RUN
WAKEUP	POC:ready	WAKEUP

Table 58.3-3 Reference to CHI Host command summary from FlexRay protocol specification

■ SUC Configuration Register 2 (SUCC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC2	R	0	0	0	0	LTN3*	LTN2*	LTN1*	LTN0*	0	0	0	LT20*	LT19*	LT18*	LT17*	LT16*
0xD084	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	LT15*	LT14*	LT13*	LT12*	LT11*	LT10*	LT9*	LT8*	LT7*	LT6*	LT5*	LT4*	LT3*	LT2*	LT1*	LT0*
	W																
Reset		0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

LT[20:0] Listen Timeout ([pdListenTimeout](#))

Configures wakeup / startup listen timeout in μ T. The range for [pdListenTimeout](#) is 1284 to 1283846 μ T.

LTN[3:0] Listen Timeout Noise ([gListenNoise](#) - 1)

Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of [pdListenTimeout](#). The range for [gListenNoise](#) is 2 to 16. **LTN[3:0]** must be configured identical in all nodes of a cluster.

Note: The wakeup / startup noise timeout is calculated as follows:

$$\text{pdListenTimeout} \cdot \text{gListenNoise} = \text{LT}[20:0] \cdot (\text{LTN}[3:0] + 1)$$

■ SUC Configuration Register 3 (SUCC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUCC3	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD088	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	WCF3*	WCF2*	WCF1*	WCF0*	WCP3*	WCP2*	WCP1*	WCP0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

WCP[3:0] Maximum Without Clock Correction Passive ([gMaxWithoutClockCorrectionPassive](#))

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.

WCF[3:0] Maximum Without Clock Correction Fatal ([gMaxWithoutClockCorrectionFatal](#))

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.

Note: The transition to HALT state is prevented if **SUCC1.HCSE** is not set.

MB91460 Series

■ NEM Configuration Register (NEMC)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NEMC	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD08C	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	NML3*	NML2*	NML1*	NML0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NML[3:0] Network Management Vector Length ([gdNetworkManagementVectorLength](#))

These bits configure the length of the NM vector. The configured length must be identical in all nodes of a cluster. Valid values are 0 to 12 bytes.

■ PRT Configuration Register 1 (PRTC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRTC1	R							0									
	W	RWP5*	RWP4*	RWP3*	RWP2*	RWP1*	RWP0*		RXW8*	RXW7*	RXW6*	RXW5*	RXW4*	RXW3*	RXW2*	RXW1*	RXW0*
0xD090																	
Reset		0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R					0	CASM6										
	W	BRP1*	BRP0*	SPP1*	SPP0*			CASM5*	CASM4*	CASM3*	CASM2*	CASM1*	CASM0*	TSST3*	TSST2*	TSST1*	TSST0*
Reset		0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1

TSST[3:0] Transmission Start Sequence Transmitter ([gdTSSTTransmitter](#))

Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 μ T = 100ns @ 10Mbps). Must be identical in all nodes of a cluster. Valid values are 3 to 15 bit times.

CASM[6:0] Collision Avoidance Symbol Max ([gdCASRxLowMax](#))

Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS). **CASM6** is fixed to '1'. Valid values are 67 to 99 bit times.

SPP[1:0] Strobe Point Position

Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by **SPP[1:0]**.

00, 11=Sample 5 (default)

01 =Sample 4

10 =Sample 6

Note:The current revision 2.1 of the FlexRay protocol requires that **SPP[1:0]** = "00". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

BRP[1:0] Baud Rate Prescaler ([gdSampleClockPeriod](#), [pSamplesPerMicrotick](#))

The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock **ERAY_SCLK** = 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

00 = 10 MBit/s (default)

[gdSampleClockPeriod](#) = 12.5 ns = 1 • **ERAY_SCLK**

[pSamplesPerMicrotick](#) = 2 (1 μT = 25 ns)

01 = 5 MBit/s

[gdSampleClockPeriod](#) = 25 ns = 2 • **ERAY_SCLK**

[pSamplesPerMicrotick](#) = 1 (1 μT = 25 ns)

10, 11 = 2.5 MBit/s

[gdSampleClockPeriod](#) = 50 ns = 4 • **ERAY_SCLK**

[pSamplesPerMicrotick](#) = 1 (1 μT = 50 ns)

RXW[8:0] Wakeup Symbol Receive Window Length ([gdWakeupSymbolRxWindow](#))

Configures the number of bit times used by the node to test the duration of the received wakeup pattern. Must be identical in all nodes of a cluster. Valid values are 76 to 301 bit times.

RWP[5:0] Repetitions of Tx Wakeup Pattern ([pWakeupPattern](#))

Configures the number of repetitions (sequences) of the Tx wakeup symbol. Valid values are 2 to 63.

■ **PRT Configuration Register 2 (PRTC2)**

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
PRTC2 0xD094	R	0	0	TXL5*	TXL4*	TXL3*	TXL2*	TXL1*	TXL0*	TXI7*	TXI6*	TXI5*	TXI4*	TXI3*	TXI2*	TXI1*	TXI0*
	W																
Reset	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	RXL5*	RXL4*	RXL3*	RXL2*	RXL1*	RXL0*	0	0	RXI5*	RXI4*	RXI3*	RXI2*	RXI1*	RXI0*
	W																
Reset	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	

RXI[5:0] Wakeup Symbol Receive Idle ([gdWakeupSymbolRxIdle](#))

Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 14 to 59 bit times.

RXL[5:0] Wakeup Symbol Receive Low ([gdWakeupSymbolRxLow](#))

Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 10 to 55 bit times.

TXI[7:0] Wakeup Symbol Transmit Idle ([gdWakeupSymbolTxIdle](#))

Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 45 to 180 bit times.

TXL[5:0] Wakeup Symbol Transmit Low ([gdWakeupSymbolTxLow](#))

Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 15 to 60 bit times.

■ **MHD Configuration Register (MHDC)**

MB91460 Series

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDC 0xD098	R	0	0	0	SLT12*	SLT11*	SLT10*	SLT9*	SLT8*	SLT7*	SLT6*	SLT5*	SLT4*	SLT3*	SLT2*	SLT1*	SLT0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	SFDL6*	SFDL5*	SFDL4*	SFDL3*	SFDL2*	SFDL1*	SFDL0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SFDL[6:0] Static Frame Data Length ([gPayloadLengthStatic](#))

Configures the cluster-wide payload length for all frames sent in the static segment in double bytes. The payload length must be identical in all nodes of a cluster. Valid values are 0 to 127.

SLT[12:0] Start of Latest Transmit ([pLatestTx](#))

Configures the maximum minislots value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if **SLT[12:0]** is set to zero. Valid values are 0 to 7981 minislots.

■ GTU Configuration Register 1 (GTUC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC1	R	0	0	0	0	0	0	0	0	0	0	0	UT19*	UT18*	UT17*	UT16*
0xD0A0	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	UT15*	UT14*	UT13*	UT12*	UT11*	UT10*	UT9*	UT8*	UT7*	UT6*	UT5*	UT4*	UT3*	UT2*	UT1*	UT0*
	W																
Reset		0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

UT[19:0] Microtick per Cycle ([pMicroPerCycle](#))

Configures the duration of the communication cycle in microticks. Valid values are 640 to 640000 μ T.

■ GTU Configuration Register 2 (GTUC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC2	R	0	0	0	0	0	0	0	0	0	0	0	SNM3*	SNM2*	SNM1*	SNM0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	MPC13*	MPC12*	MPC11*	MPC10*	MPC9*	MPC8*	MPC7*	MPC6*	MPC5*	MPC4*	MPC3*	MPC2*	MPC1*	MPC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	

MPC[13:0] Macrotick Per Cycle ([gMacroPerCycle](#))

Configures the duration of one communication cycle in macroticks. The cycle length must be identical in all nodes of a cluster. Valid values are 10 to 16000 MT.

SNM[3:0] Sync Node Max ([gSyncNodeMax](#))

Maximum number of frames within a cluster with sync frame indicator bit **SYN** set to '1'. Must be identical in all nodes of a cluster. Valid values are 2 to 15.

MB91460 Series

■ GTU Configuration Register 3 (GTUC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC3	R	0							0							
	W		MIOB6*	MIOB5*	MIOB4*	MIOB3*	MIOB2*	MIOB1*	MIOB0*		MIOA6*	MIOA5*	MIOA4*	MIOA3*	MIOA2*	MIOA1*
Reset		0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R															
	W	UIOB7*	UIOB6*	UIOB5*	UIOB4*	UIOB3*	UIOB2*	UIOB1*	UIOB0*	UIOA7*	UIOA6*	UIOA5*	UIOA4*	UIOA3*	UIOA2*	UIOA1*
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UIOA[7:0] Microtick Initial Offset Channel A ([pMicroInitialOffset\[A\]](#))

Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on [pDelayCompensation\[A\]](#) and therefore has to be set for each channel independently. Valid values are 0 to 240 μ T.

UIOB[7:0] Microtick Initial Offset Channel B ([pMicroInitialOffset\[B\]](#))

Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on [pDelayCompensation\[B\]](#) and therefore has to be set for each channel independently. Valid values are 0 to 240 μ T.

MIOA[6:0] Macrotick Initial Offset Channel A ([pMacroInitialOffset\[A\]](#))

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.

MIOB[6:0] Macrotick Initial Offset Channel B ([pMacroInitialOffset\[B\]](#))

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.

■ GTU Configuration Register 4 (GTUC4)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only. For details about configuration of **NIT[13:0]** and **OCS[13:0]** see Section "Configuration of NIT Start and Offset Correction Start" on P. 1341 .

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
GTUC4	R	0	0	OCS13*	OCS12*	OCS11*	OCS10*	OCS9*	OCS8*	OCS7*	OCS6*	OCS5*	OCS4*	OCS3*	OCS2*	OCS1*	OCS0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	NIT13*	NIT12*	NIT11*	NIT10*	NIT9*	NIT8*	NIT7*	NIT6*	NIT5*	NIT4*	NIT3*	NIT2*	NIT1*	NIT0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

NIT[13:0] Network Idle Time Start ([gMacroPerCycle](#) - [gdNIT](#) - 1)

Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if $\text{MacroTick} = \text{gMacroPerCycle} - \text{gdNIT} - 1$ and the increment pulse of MacroTick is set. Must be identical in all nodes of a cluster. Valid values are 7 to 15997 MT.

OCS[13:0] Offset Correction Start ([gOffsetCorrectionStart](#) - 1)

Determines the start of the offset correction within the NIT phase, calculated from start of cycle. Must be identical in all nodes of a cluster. For cluster consisting of E-Ray implementations only, it is sufficient to program $\text{OCS} = \text{NIT} + 1$. Valid values are 8 to 15998 MT.

■ GTU Configuration Register 5 (GTUC5)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC5 0xD0B0	R	DEC7*	DEC6*	DEC5*	DEC4*	DEC3*	DEC2*	DEC1*	DEC0*	0	0	0	CDD4*	CDD3*	CDD2*	CDD1*	CDD0*
	W																
Reset		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	DCB7*	DCB6*	DCB5*	DCB4*	DCB3*	DCB2*	DCB1*	DCB0*	DCA7*	DCA6*	DCA5*	DCA4*	DCA3*	DCA2*	DCA1*	DCA0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCA[7:0] Delay Compensation Channel A ([pDelayCompensation\[A\]](#))

Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to [cPropagationDelayMax](#) for microticks in the range of 0.0125 to 0.05μs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

Valid values are 0 to 200 μT.

DCB[7:0] Delay Compensation Channel B ([pDelayCompensation\[B\]](#))

MB91460 Series

Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to [cPropagationDelayMax](#) for microticks in the range of 0.0125 to 0.05 μ s. In practice, the minimum of the propagation delays of all sync nodes should be applied.

Valid values are 0 to 200 μ T.

CDD[4:0] Cluster Drift Damping ([pClusterDriftDamping](#))

Configures the cluster drift damping value used in clock synchronization to minimize accumulation of rounding errors. Valid values are 0 to 20 μ T.

DEC[7:0] Decoding Correction ([pDecodingCorrection](#))

Configures the decoding correction value used to determine the primary time reference point. Valid values are 14 to 143 μ T.

■ GTU Configuration Register 6 (GTUC6)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC6 0xD0B4	R	0	0	0	0	0	MOD10*	MOD9*	MOD8*	MOD7*	MOD6*	MOD5*	MOD4*	MOD3*	MOD2*	MOD1*	MOD0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ASR10*	ASR9*	ASR8*	ASR7*	ASR6*	ASR5*	ASR4*	ASR3*	ASR2*	ASR1*	ASR0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASR[10:0] Accepted Startup Range ([pdAcceptedStartupRange](#))

Number of microticks constituting the expanded range of measured deviation for startup frames during integration. Valid values are 0 to 1875 μ T.

MOD[10:0] Maximum Oscillator Drift ([pdMaxDrift](#))

Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in μ T. Valid values are 2 to 1923 μ T.

■ GTU Configuration Register 7 (GTUC7)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC7 0xD0B8	R	0	0	0	0	0	NSS9*	NSS8*	NSS7*	NSS6*	NSS5*	NSS4*	NSS3*	NSS2*	NSS1*	NSS0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SSL9*	SSL8*	SSL7*	SSL6*	SSL5*	SSL4*	SSL3*	SSL2*	SSL1*	SSL0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

SSL[9:0] Static Slot Length ([gdStaticSlot](#))

Configures the duration of a static slot in macroticks. The static slot length must be identical in all nodes of a cluster. Valid values are 4 to 659 MT.

NSS[9:0] Number of Static Slots ([gNumberOfStaticSlots](#))

Configures the number of static slots in a cycle. At least 2 coldstart nodes must be configured to startup a FlexRay network. The number of static slots must be identical in all nodes of a cluster. Valid values are 2 to 1023.

■ GTU Configuration Register 8 (GTUC8)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
GTUC8 0xD0BC	R	0	0	0	NMS12*	NMS11*	NMS10*	NMS9*	NMS8*	NMS7*	NMS6*	NMS5*	NMS4*	NMS3*	NMS2*	NMS1*	NMS0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	MSL5*	MSL4*	MSL3*	MSL2*	MSL1*	MSL0*
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

MSL[5:0] Minislot Length ([gdMinislot](#))

Configures the duration of a minislot in macroticks. The minislot length must be identical in all nodes of a cluster. Valid values are 2 to 63 MT.

NMS[12:0] Number of Minislots ([gNumberOfMinislots](#))

Configures the number of minislots within the dynamic segment of a cycle. The number of minislots must be identical in all nodes of a cluster. Valid values are 0 to 7986.

MB91460 Series

■ GTU Configuration Register 9 (GTUC9)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC9	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSI1*	DSI0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MAPO 4*	MAPO 3*	MAPO 2*	MAPO 1*	MAPO 0*	0	0	APO5*	APO4*	APO3*	APO2*	APO1*	APO0*
	W																
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

APO[5:0] Action Point Offset ([gdActionPointOffset](#))

Configures the action point offset in macroticks within static slots and symbol window. Must be identical in all nodes of a cluster. Valid values are 1 to 63 MT.

MAPO[4:0] Minislot Action Point Offset ([gdMinislotActionPointOffset](#))

Configures the action point offset in macroticks within the minislots of the dynamic segment. Must be identical in all nodes of a cluster. Valid values are 1 to 31 MT.

DSI[1:0] Dynamic Slot Idle Phase ([gdDynamicSlotIdlePhase](#))

The duration of the dynamic slot idle phase has to be greater or equal than the idle detection time. Must be identical in all nodes of a cluster. Valid values are 0 to 2 Minislot.

■ GTU Configuration Register 10 (GTUC10)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GTUC10 0xD0C4	R	0	0	0	0	0	MRC 10*	MRC9*	MRC8*	MRC7*	MRC6*	MRC5*	MRC4*	MRC3*	MRC2*	MRC1*	MRC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MOC 13*	MOC 12*	MOC 11*	MOC 10*	MOC9*	MOC8*	MOC7*	MOC6*	MOC5*	MOC4*	MOC3*	MOC2*	MOC1*	MOC0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

MOC[13:0] Maximum Offset Correction ([pOffsetCorrectionOut](#))

Holds the maximum permitted offset correction value to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value. Valid values are 5 to 15266 μ T.

MRC[10:0] Maximum Rate Correction ([pRateCorrectionOut](#))

Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value). Valid values are 2 to 1923 μ T.

■ GTU Configuration Register 11 (GTUC11)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
GTUC11	R	0	0	0	0	0	ERC2*	ERC1*	ERC0*	0	0	0	0	0	EOC2*	EOC1*	EOC0*
0xD0C8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	ERCC1	ERCC0	0	0	0	0	0	0	EOCC1	EOCC0
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EOCC[1:0] External Offset Correction Control ([vExternOffsetControl](#))

By writing to **EOCC[1:0]** the external offset correction is enabled as specified below. Should be modified only outside NIT.

00, 01 =No external offset correction

10 =External offset correction value subtracted from calculated offset correction value

11 =External offset correction value added to calculated offset correction value

ERCC[1:0] External Rate Correction Control ([vExternRateControl](#))

By writing to **ERCC[1:0]** the external rate correction is enabled as specified below. Should be modified only outside NIT.

00, 01 =No external rate correction

10 =External rate correction value subtracted from calculated rate correction value

11 =External rate correction value added to calculated rate correction value

EOC[2:0] External Offset Correction ([pExternOffsetCorrection](#))

Holds the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated offset correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 μ T.

ERC[2:0] External Rate Correction ([pExternRateCorrection](#))

Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated rate correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 μ T.

MB91460 Series

58.3.6 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses). The status vector may change faster than the Host can poll the status vector, depending on **CLKB** frequency.

■ CC Status Vector (CCSV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCSV	R	0	0	PSL5	PSL4	PSL3	PSL2	PSL1	PSL0	RCA4	RCA3	RCA2	RCA1	RCA0	WSV2	WSV1	WSV0
0xD100	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	CSI	CSAI	CSNI	0	0	SLM1	SLM0	HRQ	FSI	POCS5	POCS4	POCS3	POCS2	POCS1	POCS0
	W																
Reset		0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

POCS[5:0] Protocol Operation Control Status

Indicates the actual state of operation of the CC Protocol Operation Control

00 0000 =DEFAULT_CONFIG state

00 0001 =READY state

00 0010 =NORMAL_ACTIVE state

00 0011 =NORMAL_PASSIVE state

00 0100 =HALT state

00 0101 =MONITOR_MODE state

00 0110...00 1110 = reserved

00 1111 =CONFIG state

Indicates the actual state of operation of the POC in the wakeup path

01 0000 =WAKEUP_STANDBY state

01 0001 =WAKEUP_LISTEN state

01 0010 =WAKEUP_SEND state

01 0011 =WAKEUP_DETECT state

01 0100...01 1111 = reserved

Indicates the actual state of operation of the POC in the startup path

10 0000 =STARTUP_PREPARE state

10 0001 =COLDSTART_LISTEN state

10 0010 =COLDSTART_COLLISION_RESOLUTION state

10 0011 =COLDSTART_CONSISTENCY_CHECK state

10 0100 =COLDSTART_GAP state

10 0101 =COLDSTART_JOIN State

10 0110 =INTEGRATION_COLDSTART_CHECK state

10 0111 =INTEGRATION_LISTEN state

10 1000 =INTEGRATION_CONSISTENCY_CHECK state

10 1001 =INITIALIZE_SCHEDULE state

10 1010 =ABORT_STARTUP state

10 1011 =STARTUP_SUCCESS state

10 1100...11 1111 = reserved

FSI Freeze Status Indicator ([vPOC!Freeze](#))

Indicates that the POC has entered the HALT state due to CHI command FREEZE or due to an error condition requiring an immediate POC halt. Reset by transition from HALT to DEFAULT_CONFIG state.

HRQ Halt Request ([vPOC!CHIHaltRequest](#))

Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle. Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

SLM[1:0] Slot Mode ([vPOC!SlotMode](#))

Indicates the actual slot mode of the POC in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE. Default is SINGLE. Changes to ALL, depending on **SUCC1.TSM**. In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL. Set to SINGLE in all other states.

00 =SINGLE

01 =reserved

10 =ALL_PENDING

11 =ALL

CSNI Coldstart Noise Indicator ([vPOC!ColdstartNoise](#))

Indicates that the cold start procedure occurred under noisy conditions. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

CSAI Coldstart Abort Indicator

Coldstart aborted. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

CSI Cold Start Inhibit ([vColdStartInhibit](#))

Indicates that the node is disabled from cold starting. The flag is set whenever the POC enters READY state due to CHI command READY. The flag has to be reset under control of the Host by CHI command ALLOW_COLDSTART (**SUCC1.CMD[3:0]** = "1001").

1 = Cold starting of node disabled

0 = Cold starting of node enabled

MB91460 Series**WSV[2:0]** Wakeup Status ([vPOC!WakeupStatus](#))

Indicates the status of the current wakeup attempt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state.

000 =UNDEFINED. Wakeup not yet executed by the CC.

001 =RECEIVED_HEADER. Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.

010 =RECEIVED_WUP. Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

011 =COLLISION_HEADER. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.

100 =COLLISION_WUP. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.

101 =COLLISION_UNKNOWN. Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.

110 =TRANSMITTED. Set when the CC has successfully completed the transmission of the wakeup pattern.

111 =reserved

RCA[4:0] Remaining Coldstart Attempts ([vRemainingColdstartAttempts](#))

Indicates the number of remaining coldstart attempts. The RUN command resets this counter to the maximum number of coldstart attempts as configured by SUCC1.CSA[4:0]. The initial value of RCA[4:0] during CONFIG and DEFAULT_CONFIG state is also SUCC1.CSA[4:0].

PSL[5:0] POC Status Log

Status of POCS[5:0] immediately before entering HALT state. Set when entering HALT state. Set to HALT when FREEZE command is applied during HALT state and FSI is not already set i.e. the HALT state was not reached by FREEZE command. Reset to "00 0000" when leaving HALT state.

■ CC Error Vector (CCEV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCEV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD104	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	PTAC4	PTAC3	PTAC2	PTAC1	PTAC0	ERRM1	ERRM0	0	0	CCFC3	CCFC2	CCFC1	CCFC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

CCFC[3:0] Clock Correction Failed Counter ([vClockCorrectionFailed](#))

The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active. The Clock Correction Failed Counter is reset to '0' at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active. The Clock Correction Failed Counter stops at 15.

ERRM[1:0] Error Mode ([vPOC!ErrorMode](#))

Indicates the actual error mode of the POC.

00 =ACTIVE (green)

01 =PASSIVE (yellow)

10 =COMM_HALT (red)

11 =reserved

PTAC[4:0] Passive to Active Count ([vAllowPassiveToActive](#))

Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when **PTAC[4:0]** equals **SUCC1.PTA[4:0] -1**.

MB91460 Series

■ Slot Counter Value (SCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCV	R	0	0	0	0	0	SCCB10	SCCB9	SCCB8	SCCB7	SCCB6	SCCB5	SCCB4	SCCB3	SCCB2	SCCB1	SCCB0
0xD110	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	SCCA10	SCCA9	SCCA8	SCCA7	SCCA6	SCCA5	SCCA4	SCCA3	SCCA2	SCCA1	SCCA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

SCCA[10:0] Slot Counter Channel A ([vSlotCounter\[A\]](#))

Current slot counter value on channel A. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.

SCCB[10:0] Slot Counter Channel B ([vSlotCounter\[B\]](#))

Current slot counter value on channel B. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.

■ Macrotick and Cycle Counter Value (MTCCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MTCCV	R	0	0	0	0	0	0	0	0	0	0	CCV5	CCV4	CCV3	CCV2	CCV1	CCV0
0xD114	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	MTV13	MTV12	MTV11	MTV10	MTV9	MTV8	MTV7	MTV6	MTV5	MTV4	MTV3	MTV2	MTV1	MTV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

MTV[13:0] Macrotick Value ([vMacrotick](#))

Current macrotick value. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 15999.

CCV[5:0] Cycle Counter Value ([vCycleCounter](#))

Current cycle counter value. The value is incremented by the CC at the start of a communication cycle. Valid values are 0 to 63.

■ Rate Correction Value (RCV)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD118	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

RCV[11:0] Rate Correction Value ([vRateCorrection](#))

Rate correction value (two's complement). Calculated internal rate correction value **before** limitation. If the RCV value exceeds the limits defined by **GTUC10.MRC[10:0]**, flag **SFS.RCLR** is set to '1'.

■ Offset Correction Value (OCV)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCV	R	0	0	0	0	0	0	0	0	0	0	0	0	0	OCV18	OCV17	OCV16
0xD11C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OCV15	OCV14	OCV13	OCV12	OCV11	OCV10	OCV9	OCV8	OCV7	OCV6	OCV5	OCV4	OCV3	OCV2	OCV1	OCV0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

OCV[18:0] Offset Correction Value ([vOffsetCorrection](#))

Offset correction value (two's complement). Calculated internal offset correction value **before** limitation. If the OCV value exceeds the limits defined by **GTUC10.MOC[13:0]**, flag **SFS.OCRL** is set to '1'.

Note: The external rate / offset correction value is added to the limited rate / offset correction value.

MB91460 Series

■ Sync Frame Status (SFS)

The maximum number of valid sync frames in a communication cycle is 15.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SFS	R	0	0	0	0	0	0	0	0	0	0	0	0	RCLR	MRCS	OCLR	MOCS
0xD120	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	VSBO3	VSBO2	VSBO1	VSBO0	VSBE3	VSBE2	VSBE1	VSBE0	VSAO3	VSAO2	VSAO1	VSAO0	VSAE3	VSAE2	VSAE1	VSAE0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

VSBE[3:0] Valid Sync Frames Channel B, **even** communication cycle

Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by **SUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each even communication cycle.

VSAO[3:0] Valid Sync Frames Channel A, **odd** communication cycle

Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by **SUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

VSBO[3:0] Valid Sync Frames Channel B, **odd** communication cycle

Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by **SUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each even communication cycle.

VSAE[3:0] Valid Sync Frames Channel A, **even** communication cycle

Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by **SUCC1.TXSY** the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Note: The bit fields above are only valid if the respective channel is assigned to the CC by **SUCC1.CCHA** or **SUCC1.CCHB**.

MOCS Missing Offset Correction Signal

The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.

1 = Missing offset correction signal

0 = Offset correction signal valid

OCLR Offset Correction Limit Reached

The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by **GTUC10.MOC[13:0]**. The flag is updated by the CC at start of offset correction phase.

1 = Offset correction limit reached

0 = Offset correction below limit

MRCS Missing Rate Correction Signal

The Missing Rate Correction flag signals to the Host, that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the CC at start of offset correction phase.

- 1 = Missing rate correction signal
- 0 = Rate correction signal valid

RCLR Rate Correction Limit Reached

The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit as defined by **GTUC10.MRC[10:0]**. The flag is updated by the CC at start of offset correction phase.

- 1 = Rate correction limit reached
- 0 = Rate correction below limit

■ Symbol Window and NIT Status (SWNIT)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWNIT	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD124	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol window related status information. Updated by the CC at the end of the symbol window for each channel. During startup the status data is not updated. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

SESA Syntax Error in Symbol Window Channel A ([vSS!SyntaxErrorA](#))

- 1 = Syntax error during symbol window detected on channel A
- 0 = No syntax error detected

SBSA Slot Boundary Violation in Symbol Window Channel A ([vSS!BViolationA](#))

- 1 = Slot boundary violation during symbol window detected on channel A
- 0 = No slot boundary violation detected

TCSA Transmission Conflict in Symbol Window Channel A ([vSS!TxConflictA](#))

- 1 = Transmission conflict in symbol window detected on channel A
- 0 = No transmission conflict detected

SESB Syntax Error in Symbol Window Channel B ([vSS!SyntaxErrorB](#))

- 1 = Syntax error during symbol window detected on channel B
- 0 = No syntax error detected

SBSB Slot Boundary Violation in Symbol Window Channel B ([vSS!BViolationB](#))

- 1 = Slot boundary violation during symbol window detected on channel B
- 0 = No slot boundary violation detected

MB91460 Series

TCSB Transmission Conflict in Symbol Window Channel B ([vSS!TxConflictB](#))

- 1 = Transmission conflict in symbol window detected on channel B
- 0 = No transmission conflict detected

MTSA MTS Received on Channel A ([vSS!ValidMTSA](#))

Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag **SIR.MTSA** is set to '1'.

- 1 = MTS symbol received on channel A
- 0 = No MTS symbol received on channel A

MTSB MTS Received on Channel B ([vSS!ValidMTSB](#))

Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag **SIR.MTSB** is set to '1'.

- 1 = MTS symbol received on channel B
- 0 = No MTS symbol received on channel B

NIT related status information. Updated by the CC at the end of the NIT for each channel:

SENA Syntax Error during NIT Channel A ([vSS!SyntaxErrorA](#))

- 1 = Syntax error during NIT detected on channel A
- 0 = No syntax error detected

SBNA Slot Boundary Violation during NIT Channel A ([vSS!BViolationA](#))

- 1 = Slot boundary violation during NIT detected on channel A
- 0 = No slot boundary violation detected

SENB Syntax Error during NIT Channel B ([vSS!SyntaxErrorB](#))

- 1 = Syntax error during NIT detected on channel B
- 0 = No syntax error detected

SBNB Slot Boundary Violation during NIT Channel B ([vSS!BViolationB](#))

- 1 = Slot boundary violation during NIT detected on channel B
- 0 = No slot boundary violation detected

■ Aggregated Channel Status (ACS)

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception. The aggregated channel status also includes status data from the symbol window and the network idle time. The status data is updated (set) after each slot and aggregated until it is reset by the Host. During startup the status data is not updated. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACS	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD128	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	SBVB	CIB	CEDB	SEDB	VFRB	0	0	0	SBVA	CIA	CEDA	SEDA	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VFRA Valid Frame Received on Channel A ([vSS!ValidFrameA](#))

One or more valid frames were received on channel A in any static or dynamic slot during the observation period.

- 1 = Valid frame(s) received on channel A
- 0 = No valid frame received

SEDA Syntax Error Detected on Channel A ([vSS!SyntaxErrorA](#))

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.

- 1 = Syntax error(s) observed on channel A
- 0 = No syntax error observed

CEDA Content Error Detected on Channel A ([vSS!ContentErrorA](#))

One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.

- 1 = Frame(s) with content error received on channel A
- 0 = No frame with content error received

CIA Communication Indicator Channel A

One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

- 1 = Valid frame(s) received on channel A in slots containing any additional communication
- 0 = No valid frame(s) received in slots containing any additional communication

SBVA Slot Boundary Violation on Channel A ([vSS!BViolationA](#))

One or more slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).

- 1 = Slot boundary violation(s) observed on channel A
- 0 = No slot boundary violation observed

MB91460 Series**VFRB** Valid Frame Received on Channel B (**vSS!ValidFrameB**)

One or more valid frames were received on channel B in any static or dynamic slot during the observation period. Reset under control of the Host.

1 = Valid frame(s) received on channel B

0 = No valid frame received

SEDB Syntax Error Detected on Channel B (**vSS!SyntaxErrorB**)

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.

1 = Syntax error(s) observed on channel B

0 = No syntax error observed

CEDB Content Error Detected on Channel B (**vSS!ContentErrorB**)

One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.

1 = Frame(s) with content error received on channel B

0 = No frame with content error received

CIB Communication Indicator Channel B

One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

1 = Valid frame(s) received on channel B in slots containing any additional communication

0 = No valid frame(s) received in slots containing any additional communication

SBVB Slot Boundary Violation on Channel B (**vSS!BViolationB**)

One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).

1 = Slot boundary violation(s) observed on channel B

0 = No slot boundary violation observed

Note: The set condition of flags **CIA** and **CIB** is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

When one of the flags **SEDB**, **CEDB**, **CIB**, **SBVB** changes from '0' to '1', interrupt flag **EIR.EDB** is set to '1'. When one of the flags **SEDA**, **CEDA**, **CIA**, **SBVA** changes from '0' to '1', interrupt flag **EIR.EDA** is set to '1'.

■ Even Sync ID [1...15] (ESIDn)

Registers ESID1 to ESID15 hold the frame IDs of the sync frames received in **even** communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register ESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register ESID1 holds the respective sync frame ID as configured in message buffer 0 and flags **RXEA**, **RXEB** are set. The value is updated during the NIT of each even communication cycle. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESIDn	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W															
0xD130 - 0xD168																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	RXEB	RXEA	0	0	0	0	EID9	EID8	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

EID[9:0] Even Sync ID ([vsSyncIDListA,B even](#))

Sync frame ID even communication cycle.

RXEA Received / Configured Even Sync ID on Channel A

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = **EID[9:0]** (ESID1 only).

1 = Sync frame received on channel A / node configured to transmit sync frames

0 = No sync frame received on channel A / node not configured to transmit sync frames

RXEB Received / Configured Even Sync ID on Channel B

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = **EID[9:0]** (ESID1 only).

1 = Sync frame received on channel B / node configured to transmit sync frames

0 = No sync frame received on channel B / node not configured to transmit sync frames

MB91460 Series

■ Odd Sync ID [1...15] (OSIDn)

Registers OSID1 to OSID15 hold the frame IDs of the sync frames received in **odd** communication cycles used for clock synchronisation up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register OSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register OSID1 holds the respective sync frame ID as configured in message buffer 0 and flags **RXOA**, **RXOB** are set. The value is updated during the NIT of each odd communication cycle. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSIDn	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W															
0xD170 - 0xD1A8																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	RXOB	RXOA	0	0	0	0	OID9	OID8	OID7	OID6	OID5	OID4	OID3	OID2	OID1	OID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

OID[9:0] Odd Sync ID ([vsSyncIDListA,B odd](#))

Sync frame ID odd communication cycle.

RXOA Received / Configured Odd Sync ID on Channel A

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = **OID[9:0]** (OSID1 only).

1 = Sync frame received on channel A / node configured to transmit sync frames

0 = No sync frame received on channel A / node not configured to transmit sync frames

RXOB Received / Configured Odd Sync ID on Channel B

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = **OID[9:0]** (OSID1 only).

1 = Sync frame received on channel B / node configured to transmit sync frames

0 = No sync frame received on channel B / node not configured to transmit sync frames

■ Network Management Vector [1...3] (NMVn)

The three network management registers hold the accrued NM vector (configurable 0 to 12 bytes). The accrued NM vector is generated by the CC by bit-wise ORing each NM vector received (valid static frames with PPI = '1') on each channel (see "Network Management" on P. 1362).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

NMVn-bytes exceeding the configured NM vector length are not valid.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NMVn	R	NM31	NM30	NM29	NM28	NM27	NM26	NM25	NM24	NM23	NM22	NM21	NM20	NM19	NM18	NM17	NM16
0xD1B0 - 0xD1B8	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	NM15	NM14	NM13	NM12	NM11	NM10	NM9	NM8	NM7	NM6	NM5	NM4	NM3	NM2	NM1	NM0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 58.3-4 shows the assignment of the received payload's data bytes to the network management vector.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word																																
NMV1	Data3								Data2								Data1								Data0							
NMV2	Data7								Data6								Data5								Data4							
NMV3	Data11								Data10								Data9								Data8							

Table 58.3-4 Assignment of data bytes to network management vector

MB91460 Series

58.3.7 Message Buffer Control Registers

■ Message RAM Configuration (MRC)

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO. The register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRC	R	0	0	0	0											
0xD300	W															
Reset		0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R															
	W	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FDB[7:0] First Dynamic Buffer

0 = No group of message buffers exclusively for the static segment configured

1...127 = Message buffers 0 to **FDB** - 1 reserved for static segment

≥128 = No dynamic message buffers configured

FFB[7:0] First Buffer of FIFO

0 = All message buffers assigned to the FIFO

1...127 = Message buffers from FFB to LCB assigned to the FIFO

≥128 = No message buffer assigned to the FIFO

LCB[7:0] Last Configured Buffer

0...127 = Number of message buffers is **LCB** + 1

≥128 = No message buffer configured

SEC[1:0] Secure Buffers

Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state. For temporary unlocking see "Host Handling of Parity Errors" on P. 1388 .

00 = Reconfiguration of message buffers enabled with numbers < **FFB** enabled

Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if **SPLM** = '1', also message buffer 1) is always locked

01 = Reconfiguration of message buffers with numbers < **FDB** and with numbers ≥ **FFB** locked and transmission of message buffers for static segment with numbers ≥ **FDB** disabled

10 = Reconfiguration of all message buffers locked

11 = Reconfiguration of all message buffers locked

and transmission of message buffers for static segment with numbers ≥ **FDB** disabled

SPLM Sync Frame Payload Multiplex

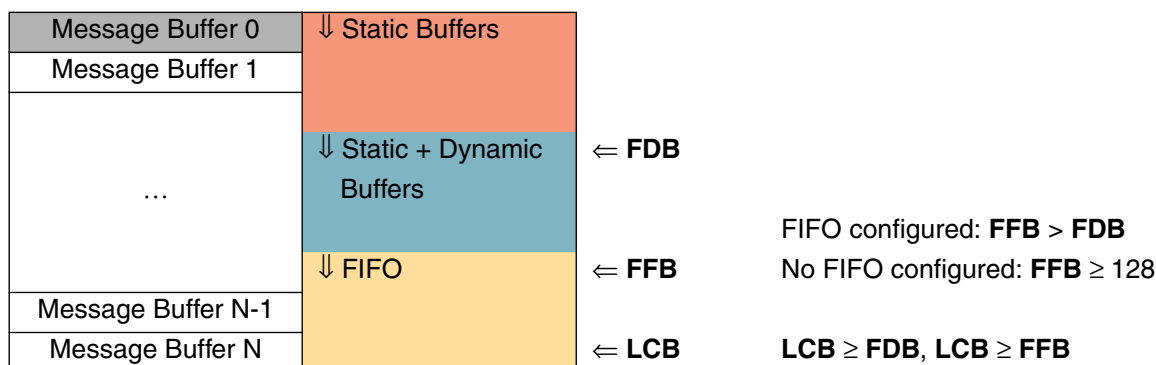
This bit is only evaluated if the node is configured as sync node (**SUCC1.TXSY** = '1') or for single slot mode operation (**SUCC1.TSM** = '1'). When this bit is set to '1' message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B. When this bit is set to '0', sync frames are transmitted from message buffer 0 with the same payload data on both

channels. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen accordingly.

1 = Both message buffers 0 and 1 are locked against reconfiguration

0 = Only message buffer 0 locked against reconfiguration

Note: In case the node is configured as sync node (**SUCC1.TXSY** = '1') or for single slot mode operation (**SUCC1.TSM** = '1'), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.



The programmer has to ensure that the configuration defined by **FDB[7:0]**, **FFB[7:0]**, and **LCB[7:0]** is valid.

The CC does not check for erroneous configurations!

Note: The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see Section 58.4.12 Message RAM (Page No.1381).

In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "Static Buffers" or at the beginning of the "Static + Dynamic Buffers" section.

The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.

The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via **WRHS2.PLC[6:0]** and **WRHS3.DP[10:0]**. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.

MB91460 Series**■ FIFO Rejection Filter (FRF)**

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO. The FRF register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRF	R	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
	W															
Reset		0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	FID10*	FID9*	FID8*	FID7*	FID6*	FID5*	FID4*	FID3*	FID2*	FID1*	FID0*	CH1*	CH0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

CH[1:0] Channel Filter

11 =no reception

10 =receive only on channel A

01 =receive only on channel B

00 =receive on both channels

Note: If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

FID[10:0] Frame ID Filter

Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When **FRFM.MFID[10:0]** is zero, a frame ID filter value of zero means that **no** frame ID is rejected.

0...2047 = Frame ID filter values

CYF[6:0] Cycle Counter Filter

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles **not** belonging to the cycle set specified by **CYF[6:0]**, all frames are rejected. For details about the configuration of the cycle counter filter see Section "[Cycle Counter Filtering](#)" on P. 1364 .

RSS Reject in Static Segment

If this bit is set, the FIFO is used only for the dynamic segment.

1 = Reject messages in static segment

0 = FIFO also used for static segment

RNF Reject Null Frames

If this bit is set, received null frames are not stored in the FIFO.

1 = Reject all null frames

0 = Null frames are stored in the FIFO

■ FIFO Rejection Filter Mask (FRFM)

The FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set, it indicates that the corresponding bit in the FRF register will not be considered for rejection filtering. The FRFM register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD308	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	MFID 10*	MFID 9*	MFID 8*	MFID 7*	MFID 6*	MFID 5*	MFID 4*	MFID 3*	MFID 2*	MFID 1*	MFID 0*	0
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MFID[10:0] Mask Frame ID Filter

1 = Ignore corresponding frame ID filter bit.

0 = Corresponding frame ID filter bit is used for rejection filtering

■ FIFO Critical Level (FCL)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCL	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD30C	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	CL7*	CL6*	CL5*	CL4*	CL3*	CL2*	CL1*	CL0*
	W															
Reset		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

CL[7:0] Critical Level

When the receive FIFO fill level **FSR.RFFL[7:0]** is equal or greater than the critical level configured by **CL[7:0]**, the receive FIFO critical level flag **FSR.RFCL** is set. If **CL[7:0]** is programmed to values > 128, bit **FSR.RFCL** is never set. When **FSR.RFCL** changes from '0' to '1' bit **SIR.RFCL** is set to '1', and if enabled, an interrupt is generated.

MB91460 Series

58.3.8 Message Buffer Status Registers

■ Message Handler Status (MHDS)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MHDS 0xD310	R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM							
	W										MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. The register will also be cleared by hard reset or by CHI command CLEAR_RAMs.

PIBF Parity Error Input Buffer RAM 1,2

- 1 = Parity error occurred when reading Input Buffer RAM 1,2
- 0 = No parity error

POBF Parity Error Output Buffer RAM 1,2

- 1 = Parity error occurred when reading Output Buffer RAM 1,2
- 0 = No parity error

PMR Parity Error Message RAM

- 1 = Parity error occurred when reading the Message RAM
- 0 = No parity error

PTBF1 Parity Error Transient Buffer RAM A

- 1 = Parity error occurred when reading Transient Buffer RAM A
- 0 = No parity error

PTBF2 Parity Error Transient Buffer RAM B

- 1 = Parity error occurred when reading Transient Buffer RAM B
- 0 = No parity error

Note: When one of the flags **PIBF**, **POBF**, **PMR**, **PTBF1**, **PTBF2** changes from '0' to '1' **EIR.PERR** is set to '1'.

FMBD Faulty Message Buffer Detected

- 1 = Message buffer referenced by **FMB[6:0]** holds faulty data due to a parity error
- 0 = No faulty message buffer

MFMB Multiple Faulty Message Buffers detected

- 1 = Another faulty message buffer was detected while flag **FMBD** is set
- 0 = No additional faulty message buffer

CRAM Clear all internal RAM's

Signals that execution of the CHI command CLEAR_RAMs is ongoing (all bits of all internal RAM blocks are written to '0'). The bit is set by hard reset or by CHI command CLEAR_RAMs.

- 1 = Execution of the CHI command CLEAR_RAMs ongoing
- 0 = No execution of the CHI command CLEAR_RAMs

FMB[6:0] Faulty Message Buffer

Parity error occurred when reading from the message buffer or when transferring data from Input Buffer or Transient Buffer 1,2 to the message buffer referenced by **FMB[6:0]**. Value only valid when one of the flags **PIBF**, **PMR**, **PTBF1**, **PTBF2**, and flag **FMBD** is set. Is not updated while flag **FMBD** is set.

MBT[6:0] Message Buffer Transmitted

Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective **TXR** flag in the **TXRQ1/2/3/4** registers was reset.

MBU[6:0] Message Buffer Updated

Number of message buffer that was updated last by the CC. For this message buffer the respective **ND** and / or **MBC** flag in the **NDAT1/2/3/4** registers and the **MBSC1/2/3/4** registers are also set.

Note: **MBT[6:0]** and **MBU[6:0]** are reset when the CC leaves CONFIG state or enters STARTUP state.

■ Last Dynamic Transmit Slot (LDTS)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LDTS	R	0	0	0	0	0	LDTB10	LDTB9	LDTB8	LDTB7	LDTB6	LDTB5	LDTB4	LDTB3	LDTB2	LDTB1	LDTB0
0xD314	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	LDTA10	LDTA9	LDTA8	LDTA7	LDTA6	LDTA5	LDTA4	LDTA3	LDTA2	LDTA1	LDTA0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The register is reset when the CC leaves CONFIG state or enters STARTUP state.

LDTA[10:0] Last Dynamic Transmission Channel A

Value of **vSlotCounter[A]** at the time of the last frame transmission on channel A in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

LDTB[10:0] Last Dynamic Transmission Channel B

Value of **vSlotCounter[B]** at the time of the last frame transmission on channel B in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

MB91460 Series**■ FIFO Status Register (FSR)**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FSR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD318	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	RFFL7	RFFL6	RFFL5	RFFL4	RFFL3	RFFL2	RFFL1	RFFL0	0	0	0	0	0	RFO	RFCL	RFNE
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Note: The register is reset when the CC leaves CONFIG state or enters STARTUP state.

RFNE Receive FIFO Not Empty

This flag is set by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag **SIR.RFNE** is set. The bit is reset after the Host has read all message from the FIFO.

1 = Receive FIFO is not empty

0 = Receive FIFO is empty

RFCL Receive FIFO Critical Level

This flag is set when the receive FIFO fill level **RFFL[7:0]** is equal or greater than the critical level as configured by **FCL.CL[7:0]**. The flag is cleared by the CC as soon as **RFFL[7:0]** drops below **FCL.CL[7:0]**. When **RFCL** changes from '0' to '1' bit **SIR.RFCL** is set to '1', and if enabled, an interrupt is generated.

1 = Receive FIFO critical level reached

0 = Receive FIFO below critical level

RFO Receive FIFO Overrun

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag **EIR.RFO** is set. The flag is cleared by the next FIFO read access issued by the Host.

1 = A receive FIFO overrun has been detected

0 = No receive FIFO overrun detected

RFFL[7:0] Receive FIFO Fill Level

Number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.

■ Message Handler Constraints Flags (MHDF)

Some constraints exist for the Message Handler regarding **CLKB** frequency, Message RAM configuration, and FlexRay bus traffic (see Addendum to E-Ray FlexRay IP-Module Specification). To simplify software development, constraints violations are reported by setting flags in the MHDF.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDF	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD31C	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0									
	W							WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hard reset will also clear the register. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

SNUA Status Not Updated Channel A

This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel A.

1 = MBS for channel A not updated

0 = No overload condition occurred when updating MBS for channel A

SNUB Status Not Updated Channel B

This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status MBS with respect to channel B.

1 = MBS for channel B not updated

0 = No overload condition occurred when updating MBS for channel B

FNFA Find Sequence Not Finished Channel A

This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel A.

1 = Find sequence not finished for channel A

0 = No find sequence not finished for channel A

FNFB Find Sequence Not Finished Channel B

This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel B.

1 = Find sequence not finished for channel B

0 = No find sequence not finished for channel B

TBFA Transient Buffer Access Failure A

This flag is set by the CC when a read or write access to TBF A requested by PRT A could not complete within the available time.

1 = TBF A access failure

0 = No TBF A access failure

MB91460 Series**TBFB** Transient Buffer Access Failure B

This flag is set by the CC when a read or write access to TBF B requested by PRT B could not complete within the available time.

1 = TBF B access failure

0 = No TBF B access failure

TNSA Transmission Not Started Channel A

This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.

1 = Transmission not started on channel A

0 = No transmission not started on channel A

TNSB Transmission Not Started Channel B

This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.

1 = Transmission not started on channel B

0 = No transmission not started on channel B

WAHP Write Attempt to Header Partition

Outside DEFAULT_CONFIG and CONFIG state this flag is set by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.

1 = Write attempt to header partition

0 = No write attempt to header partition

Note: When one of the flags **SNUA**, **SNUB**, **FNFA**, **FNFB**, **TBFA**, **TBFB**, **TNSA**, **TNSB**, **WAHP** changes from '0' to '1', interrupt flag **EIR.MHF** is set to '1'.

■ Transmission Request 1/2/3/4 (TXRQ1/2/3/4)

The four registers reflect the state of the **TXR** flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining **TXR** flags have no meaning.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
TXRQ4	R	TXR127	TXR126	TXR125	TXR124	TXR123	TXR122	TXR121	TXR120	TXR119	TXR118	TXR117	TXR116	TXR115	TXR114	TXR113	TXR112
	W																
0xD32C																	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR111	TXR110	TXR109	TXR108	TXR107	TXR106	TXR105	TXR104	TXR103	TXR102	TXR101	TXR100	TXR99	TXR98	TXR97	TXR96
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
TXRQ3	R	TXR95	TXR94	TXR93	TXR92	TXR91	TXR90	TXR89	TXR88	TXR87	TXR86	TXR85	TXR84	TXR83	TXR82	TXR81	TXR80
0xD328	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR79	TXR78	TXR77	TXR76	TXR75	TXR74	TXR73	TXR72	TXR71	TXR70	TXR69	TXR68	TXR67	TXR66	TXR65	TXR64
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
TXRQ2	R	TXR63	TXR62	TXR61	TXR60	TXR59	TXR58	TXR57	TXR56	TXR55	TXR54	TXR53	TXR52	TXR51	TXR50	TXR49	TXR48
0xD324	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXR47	TXR46	TXR45	TXR44	TXR43	TXR42	TXR41	TXR40	TXR39	TXR38	TXR37	TXR36	TXR35	TXR34	TXR33	TXR32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
TXRQ1	R	TXR31	TXR30	TXR29	TXR28	TXR27	TXR26	TXR25	TXR24	TXR23	TXR22	TXR21	TXR20	TXR19	TXR18	TXR17	TXR16
0xD320	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	TXR15	TXR14	TXR13	TXR12	TXR11	TXR10	TXR9	TXR8	TXR7	TXR6	TXR5	TXR4	TXR3	TXR2	TXR1	TXR0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TXR[127:0] Transmission Request

If the flag is set, the respective message buffer is ready for transmission respectively transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

MB91460 Series

■ New Data 1/2/3/4 (NDAT1/2/3/4)

The four registers reflect the state of the **ND** flags of all configured message buffers. **ND** flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining **ND** flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT4 0xD33C	R	ND127	ND126	ND125	ND124	ND123	ND122	ND121	ND120	ND119	ND118	ND117	ND116	ND115	ND114	ND113	ND112
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND111	ND110	ND109	ND108	ND107	ND106	ND105	ND104	ND103	ND102	ND101	ND100	ND99	ND98	ND97	ND96
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT3 0xD338	R	ND95	ND94	ND93	ND92	ND91	ND90	ND89	ND88	ND87	ND86	ND85	ND84	ND83	ND82	ND81	ND80
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND79	ND78	ND77	ND76	ND75	ND74	ND73	ND72	ND71	ND70	ND69	ND68	ND67	ND66	ND65	ND64
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT2 0xD334	R	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NDAT1 0xD330	R	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ND[127:0] New Data

The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An **ND** flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

■ Message Buffer Status Changed 1/2/3/4 (MBSC1/2/3/4)

The four registers reflect the state of the **MBC** flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining **MBC** flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MBSC4	R	MBC127	MBC126	MBC125	MBC124	MBC123	MBC122	MBC121	MBC120	MBC119	MBC118	MBC117	MBC116	MBC115	MBC114	MBC113	MBC112
0xD34C	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBC111	MBC110	MBC109	MBC108	MBC107	MBC106	MBC105	MBC104	MBC103	MBC102	MBC101	MBC100	MBC99	MBC98	MBC97	MBC96
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MBSC3	R	MBC95	MBC94	MBC93	MBC92	MBC91	MBC90	MBC89	MBC88	MBC87	MBC86	MBC85	MBC84	MBC83	MBC82	MBC81	MBC80
0xD348	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBC79	MBC78	MBC77	MBC76	MBC75	MBC74	MBC73	MBC72	MBC71	MBC70	MBC69	MBC68	MBC67	MBC66	MBC65	MBC64
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
MBSC2	R	MBC63	MBC62	MBC61	MBC60	MBC59	MBC58	MBC57	MBC56	MBC55	MBC54	MBC53	MBC52	MBC51	MBC50	MBC49	MBC48
0xD344	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBC47	MBC46	MBC45	MBC44	MBC43	MBC42	MBC41	MBC40	MBC39	MBC38	MBC37	MBC36	MBC35	MBC34	MBC33	MBC32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBSC1	R	MBC31	MBC30	MBC29	MBC28	MBC27	MBC26	MBC25	MBC24	MBC23	MBC22	MBC21	MBC20	MBC19	MBC18	MBC17	MBC16
0xD340	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	MBC15	MBC14	MBC13	MBC12	MBC11	MBC10	MBC9	MBC8	MBC7	MBC6	MBC5	MBC4	MBC3	MBC2	MBC1	MBC0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MBC[127:0] Message Buffer Status Changed

An **MBC** flag is set whenever the Message Handler changes one of the status flags **VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB** in the header section (see "[Message Buffer Status \(MBS\)](#)" on P. 1333 and "[Header Partition](#)" on P. 1382 , header 4) of the respective message buffer. An **MBC** flag is reset when the header section of

MB91460 Series

the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

58.3.9 Identification Registers

■ Core Release Register (CREL)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CREL	R	REL3	REL2	REL1	REL0	STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	STEP0	YEAR3	YEAR2	YEAR1	YEAR0
0xD3F0	W																
Reset	release info																

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MON7	MON6	MON5	MON4	MON3	MON2	MON1	MON0	DAY7	DAY6	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
	W																
Reset	release info																

DAY[7:0] Design Time Stamp, Day
Two digits, BCD-coded.

MON[7:0]. Design Time Stamp, Month
Two digits, BCD-coded.

YEAR[3:0]. Design Time Stamp, Year
One digit, BCD-coded.

STEP[7:0]. Step of Core Release
Two digits, BCD-coded.

REL[3:0] Core Release
One digit, BCD-coded.

Table 58.3-5 below shows how releases are coded in register CREL.

Release	Step	Sub-Step	Name
0	7	0	Beta2
0	7	1	Beta2ct
0	7	2	Revision 1.0RC1
1	0	0	Revision 1.0.0
1	0	1	Revision 1.0.1
1	0	2	Revision 1.0.2

Table 58.3-5 Coding for releases

■ Endian Register (ENDN)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENDN	R	ETV31	ETV30	ETV29	ETV28	ETV27	ETV26	ETV25	ETV24	ETV23	ETV22	ETV21	ETV20	ETV19	ETV18	ETV17	ETV16
0xD3F4	W																
Reset		1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	1

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ETV15	ETV14	ETV13	ETV12	ETV11	ETV10	ETV9	ETV8	ETV7	ETV6	ETV5	ETV4	ETV3	ETV2	ETV1	ETV0
	W																
Reset		0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1

ETV[31:0] Endianness Test Value

The endianness test value is 0x87654321.

58.3.10 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in Section "[Message Buffer Status \(MBS\)](#)" on P. 1333 is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via **WRHS2.PLC[6:0]** and **WRHS3.DP[10:0]**. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in Section "[Data Transfer from Input Buffer to Message RAM](#)" on P. 1375 .

■ Write Data Section [1...64] (WRDSn)

Holds the data words to be transferred to the data section of the addressed message buffer. The data words (DW_n) are written to the Message RAM in transmission order from DW_1 (byte0, byte1) to DW_{PL} (PL = number of data words as defined by the payload length configured **WRHS2.PLC[6:0]**).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRDSn	R															
	W	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R															
	W	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD[31:0] Message Data

MD[7:0]= DW_{2n-1} , byte_{4n-4}

MD[15:8]= DW_{2n-1} , byte_{4n-3}

MD[23:16]= DW_{2n} , byte_{4n-2}

MD[31:24]= DW_{2n} , byte_{4n-1}

MB91460 Series

Note: DW127 is located on WRDS64.MD[15:0]. In this case WRDS64.MD[31:16] is unused (no valid data). The Input Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMs.

Transmission order on the FlexRay bus is WRDSn[7:0], WRDSn[15:8], WRDSn[23:16], WRDSn[31:24] with the most significant bit transmitted first. To check how the E-Ray's endianness matches with the Host CPU's endianness, read register ENDN.

■ Write Header Section 1 (WRHS1)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
WRHS1 0xD500	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

FID[10:0] Frame ID

Frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message. **Message buffers with frame ID = '0' are considered as not valid.**

CYC[6:0] Cycle Code

The 7-bit cycle code determines the cycle set used for cycle counter filtering. For details about the configuration of the cycle code see Section ["Cycle Counter Filtering" on P. 1364](#).

CHA, CHB Channel Filter Control

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

CHA	CHB	Transmit Buffer transmit frame on	Receive Buffer store frame received from
1	1	both channels (static segment only)	channel A or B (store first semantically valid frame, static segment only)
1	0	channel A	channel A
0	1	channel B	channel B
0	0	no transmission	ignore frame

Note: If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no frames are transmitted resp. received frames are ignored (same function as **CHA = CHB = '0'**)

CFG Message Buffer Direction Configuration Bit

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated.

1 = The corresponding buffer is configured as **Transmit Buffer**

0 = The corresponding buffer is configured as **Receive Buffer**

PPIT Payload Preamble Indicator Transmit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames. If the bit is set in a static message buffer, the respective message buffer holds network management information. If the bit is set in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the E-Ray module, but can be done by the Host.

- 1 = Payload Preamble Indicator set
- 0 = Payload Preamble Indicator not set

TXM Transmission Mode

This bit is used to select the transmission mode (see Section "Transmit Buffers" on P. 1366).

- 1 = Single-shot mode
- 0 = Continuous mode

MBI Message Buffer Interrupt

This bit enables the receive / transmit interrupt for the corresponding message buffer. After a dedicated receive buffer has been updated by the Message Handler, flag **SIR.RXI** and /or **SIR.MBSI** are set. After a transmission has completed flag **SIR.TXI** is set.

- 1 = The corresponding message buffer interrupt is enabled
- 0 = The corresponding message buffer interrupt is disabled

■ **Write Header Section 2 (WRHS2)**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRHS2 0xD504	R	0	0	0	0	0	0	0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC[10:0] Header CRC (vRF!Header!HeaderCRC)

Receive Buffer: Configuration not required

Transmit Buffer: Header CRC calculated and configured by the Host

For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by **MHDC.SFDL[6:0]**.

PLC[6:0] Payload Length Configured

Length of data section (number of 2-byte words) as configured by the Host. During static segment the static frame payload length as configured by **MHDC.SFDL[6:0]** defines the payload length for all static frames. If the payload length configured by **PLC[6:0]** is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is logical zero (see also Section "Transmit Buffers" on P. 1366).

MB91460 Series**■ Write Header Section 3 (WRHS3)**

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRHS3	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD508	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DP[10:0] Data Pointer

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

■ Input Buffer Command Mask (IBCM)

Configures how the message buffer in the Message RAM selected by register IBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits **LHSH**, **LDSH**, and **STXRH** are swapped with bits **LHSS**, **LDSS**, and **STXRS** to keep them attached to the respective Input Buffer transfer.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRS	LDSS	LHSS
0xD510	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRH	LDSH	LHSH
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LHSH Load Header Section Host

- 1 = Header section selected for transfer from Input Buffer to the Message RAM
- 0 = Header section is not updated

LDSH Load Data Section Host

- 1 = Data section selected for transfer from Input Buffer to the Message RAM
- 0 = Data section is not updated

STXRH Set Transmission Request Host

If this bit is set to '1', the **TXR** flag for the selected message buffer is set in the TXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed. **TXR** is evaluated for transmit buffers only.

- 1 = Set **TXR** flag, transmit buffer released for transmission
- 0 = Reset **TXR** flag

LHSS Load Header Section Shadow

- 1 = Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
- 0 = Header section is not updated

LDSS Load Data Section Shadow

- 1 = Data section selected for transfer from Input Buffer to the Message RAM
(transfer ongoing or finished)
- 0 = Data section is not updated

STXRS Set Transmission Request Shadow

- 1 = Set **TXR** flag, transmit buffer released for transmission (operation ongoing or finished)
- 0 = Reset **TXR** flag

■ Input Buffer Command Request (IBCR)

When the Host writes the number of the target message buffer in the Message RAM to **IBRH[6:0]**, IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under **IBRH[6:0]** and **IBRS[6:0]** are also swapped (see also Section "Data Transfer from Input Buffer to Message RAM" on P. 1375).

With this write operation the **IBSYS** is set to '1'. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by **IBRS[6:0]**.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, **IBSYS** is set back to '0' and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to **IBRH[6:0]**.

If a write access to **IBRH[6:0]** occurs while **IBSYS** is '1', **IBSYH** is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, **IBSYH** is reset to '0'. **IBSYS** remains set to '1', and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under **IBRH[6:0]** and **IBRS[6:0]** are also swapped.

Any write access to an Input Buffer register while both **IBSYS** and **IBSYH** are set will cause the error flag **EIR.IBA** to be set. In this case the Input Buffer will not be changed.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBCR	R	IBSYS	0	0	0	0	0	0	0	0	IBRS6	IBRS5	IBRS4	IBRS3	IBRS2	IBRS1	IBRS0
	W																
0xD514																	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	IBSYH	0	0	0	0	0	0	0	0	IBRH6	IBRH5	IBRH4	IBRH3	IBRH2	IBRH1	IBRH0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IBRH[6:0] Input Buffer Request Host

Selects the target message buffer in the Message RAM for data transfer from Input Buffer.

Valid values are 0x00 to 0x7F (0...127).

IBSYH Input Buffer Busy Host

Set to '1' by writing **IBRH[6:0]** while **IBSYS** is still '1'. After the ongoing transfer between IBF Shadow and the Message RAM has completed, the **IBSYH** is set back to '0'.

- 1 = Request while transfer between IBF Shadow and Message RAM in progress
- 0 = No request pending

IBRS[6:0] Input Buffer Request Shadow

Number of the target message buffer actually updated / lately updated.

Valid values are 0x00 to 0x7F (0...127).

MB91460 Series

IBSYS Input Buffer Busy Shadow

Set to '1' after writing **IBRH[6:0]**. When the transfer between IBF Shadow and the Message RAM has completed, **IBSYS** is set back to '0'.

1 = Transfer between IBF Shadow and Message RAM in progress

0 = Transfer between IBF Shadow and Message RAM completed

58.3.11 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in Section ["Data Transfer from Message RAM to Output Buffer" on P. 1377](#).

■ Read Data Section [1...64] (RDDS_n)

Holds the data words read from the data section of the addressed message buffer. The data words (DW_n) are read from the Message RAM in reception order from DW₁ (byte0, byte1) to DW_{PL} (PL = number of data words as defined by the payload length configured **RDHS2.PLC[6:0]**).

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDDS _n	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
	W																
0xD600 - 0xD6FC																	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MD[31:0] Message Data

MD[7:0] = DW_{2n-1}, byte_{4n-4}

MD[15:8] = DW_{2n-1}, byte_{4n-3}

MD[23:16] = DW_{2n}, byte_{4n-2}

MD[31:24] = DW_{2n}, byte_{4n-1}

Note: DW127 is located on RDDS64.MD[15:0]. In this case RDDS64.MD[31:16] is unused (no valid data). The Output Buffer RAMs are initialized to zero when leaving hard reset or by CHI command CLEAR_RAMs.

■ Read Header Section 1 (RDHS1)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RDHS1	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
0xD700	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Values as configured by the Host via WRHS1:

FID[10:0] Frame ID

CYC[6:0] Cycle Code

CHA, CHB Channel Filter Control

CFG Message Buffer Direction Configuration Bit

PPIT Payload Preamble Indicator Transmit

TXM Transmission Mode

MBI Message Buffer Interrupt

In case that the message buffer read from the Message RAM belongs to the receive FIFO, **FID[10:0]** holds the received frame ID, while **CYC[6:0]**, **CHA**, **CHB**, **CFG**, **PPIT**, **TXM**, and **MBI** are reset to '0'.

MB91460 Series

■ Read Header Section 2 (RDHS2)

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDHS2	R	0	PLR6	PLR5	PLR4	PLR3	PLR2	PLR1	PLR0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
0xD704	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC[10:0] Header CRC ([vRF!Header!HeaderCRC](#))

Receive Buffer: Header CRC updated from received data frames

Transmit Buffer: Header CRC calculated and configured by the Host

PLC[6:0] Payload Length Configured

Length of data section (number of 2-byte words) as configured by the Host.

PLR[6:0] Payload Length Received ([vRF!Header!Length](#))

Payload length value updated from received data frames (exception: if message buffer belongs to the receive FIFO **PLR[6:0]** is also updated from received null frames)

When a message is stored into a message buffer the following behaviour with respect to payload length received and payload length configured is implemented:

PLR[6:0] > PLC[6:0]: The payload data stored in the message buffer is truncated to the payload length configured if **PLC[6:0]** even or else truncated to **PLC[6:0] + 1**.

PLR[6:0] ≤ PLC[6:0]: The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by **PLC[6:0]** are filled with undefined data

PLR[6:0] = zero: The message buffer's data section is filled with undefined data

PLC[6:0] = zero: Message buffer has no data section configured. No data is stored into the message buffer's data section.

Note: The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is **PLC[6:0]** rounded to the next even value. **PLC[6:0]** should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.

■ Read Header Section 3 (RDHS3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RDHS3	R	0	0	RES	PPI	NFI	SYN	SFI	RCI	0	0	RCC5	RCC4	RCC3	RCC2	RCC1	RCC0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DP[10:0] Data Pointer

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

RCC[5:0] Receive Cycle Count ([vRF!Header!CycleCount](#))

Cycle counter value updated from received data frame.

RCI Received on Channel Indicator ([vSS!Channel](#))

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

1 = Frame received on channel A

0 = Frame received on channel B

SFI Startup Frame Indicator ([vRF!Header!SuFIndicator](#))

A startup frame is marked by the startup frame indicator.

1 = The received frame is a startup frame

0 = The received frame is not a startup frame

SYN Sync Frame Indicator ([vRF!Header!SyFIndicator](#))

A sync frame is marked by the sync frame indicator.

1 = The received frame is a sync frame

0 = The received frame is not a sync frame

NFI Null Frame Indicator ([vRF!Header!NFIndicator](#))

Is set to '1' after storage of the first received data frame.

1 = At least one data frame has been stored into the respective message buffer

0 = Up to now no data frame has been stored into the respective message buffer

PPI Payload Preamble Indicator ([vRF!Header!PPIndicator](#))

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

1 = Static segment: Network management vector in the first part of the payload

Dynamic segment: Message ID in the first part of the payload

0 = The payload segment of the received frame does not contain a network management vector nor a message ID

RES Reserved Bit ([vRF!Header!Reserved](#))

Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

Note: Header 3 is updated from data frames only.

MB91460 Series**■ Message Buffer Status (MBS)**

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer. The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state. If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated. The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all MBS flags are reset to zero independent of which IBCM bits are set or not. For details about receive / transmit filtering see Sections 58.4.7 Filtering and Masking (Page No.1363), 58.4.8 Transmit Process (Page No.1366), and 58.4.9 Receive Process (Page No.1368). Whenever the Message Handler changes one of the flags **VFRA**, **VFRB**, **SEOA**, **SEOB**, **CEOA**, **CEOB**, **SVOA**, **SVOB**, **TCIA**, **TCIB**, **ESA**, **ESB**, **MLST**, **FTA**, **FTB** the respective message buffer's **MBC** flag in registers MBSC1/2/3/4 is set.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBS	R	0	0	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	0	0	CCS5	CCS4	CCS3	CCS2	CCS1	CCS0
0xD70C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FTB	FTA	0	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VFRA Valid Frame Received on Channel A ([vSS!ValidFrameA](#))

A valid frame indication is set if a valid frame was received on channel A.

1 = Valid frame received on channel A

0 = No valid frame received on channel A

VFRB Valid Frame Received on Channel B ([vSS!ValidFrameB](#))

A valid frame indication is set if a valid frame was received on channel B.

1 = Valid frame received on channel B

0 = No valid frame received on channel B

SEOA Syntax Error Observed on Channel A ([vSS!SyntaxErrorA](#))

A syntax error was observed in the assigned slot on channel A.

1 = Syntax error observed on channel A

0 = No syntax error observed on channel A

SEOB Syntax Error Observed on Channel B ([vSS!SyntaxErrorB](#))

A syntax error was observed in the assigned slot on channel B.

1 = Syntax error observed on channel B

0 = No syntax error observed on channel B

CEOA Content Error Observed on Channel A ([vSS!ContentErrorA](#))

A content error was observed in the assigned slot on channel A.

1 = Content error observed on channel A

0 = No content error observed on channel A

CEOB Content Error Observed on Channel B ([vSS!ContentErrorB](#))

A content error was observed in the assigned slot on channel B.

- 1 = Content error observed on channel B
- 0 = No content error observed on channel B

SVOA Slot Boundary Violation Observed on Channel A ([vSS!BViolationA](#))

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A.

- 1 = Slot boundary violation observed on channel A
- 0 = No slot boundary violation observed on channel A

SVOB Slot Boundary Violation Observed on Channel B ([vSS!BViolationB](#))

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B.

- 1 = Slot boundary violation observed on channel B
- 0 = No slot boundary violation observed on channel B

TCIA Transmission Conflict Indication Channel A ([vSS!TxConflictA](#))

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

- 1 = Transmission conflict occurred on channel A
- 0 = No transmission conflict occurred on channel A

TCIB Transmission Conflict Indication Channel B ([vSS!TxConflictB](#))

A transmission conflict indication is set if a transmission conflict has occurred on channel B.

- 1 = Transmission conflict occurred on channel B
- 0 = No transmission conflict occurred on channel B

ESA Empty Slot Channel A

In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots.

- 1 = No bus activity detected in the assigned slot on channel A
- 0 = Bus activity detected in the assigned slot on channel A

ESB Empty Slot Channel B

In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots.

- 1 = No bus activity detected in the assigned slot on channel B
- 0 = Bus activity detected in the assigned slot on channel B

MLST Message Lost

The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame. Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset by a Host write to the message buffer via IBF or when a new message is stored into the message buffer **after** the message buffers **ND** flag was reset by reading out the message buffer via OBF.

- 1 = Unprocessed message was overwritten
- 0 = No message lost

FTA Frame Transmitted on Channel A

Indicates that this node has transmitted a data frame in the configured slot on channel A.

- 1 = Data frame transmitted on channel A
- 0 = No data frame transmitted on channel A

MB91460 Series**FTB** Frame Transmitted on Channel B

Indicates that this node has transmitted a data frame in the configured slot on channel B.

1 = Data frame transmitted on channel B

0 = No data frame transmitted on channel B

Note: The FlexRay protocol specification requires that **FTA**, and **FTB** can only be reset by the Host. Therefore the Cycle Count Status **CCS[5:0]** for these bits is only valid for the cycle where the bits are set to '1'.

CCS[5:0] Cycle Count Status

Actual cycle count when status was updated.

For receive buffers (**CFG** = '0') the following status bits are updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.

RCIS Received on Channel Indicator Status (**vSS!Channel**)

Indicates the channel on which the frame was received.

1 = Frame received on channel A

0 = Frame received on channel B

SFIS Startup Frame Indicator Status (**vRF!Header!SuFIndicator**)

A startup frame is marked by the startup frame indicator.

1 = The received frame is a startup frame

0 = No startup frame received

SYNS Sync Frame Indicator Status (**vRF!Header!SyFIndicator**)

A sync frame is marked by the sync frame indicator.

1 = The received frame is a sync frame

0 = No sync frame received

NFIS Null Frame Indicator Status (**vRF!Header!NFIndicator**)

If set to '0' the payload segment of the received frame contains no usable data.

1 = Received frame is **not** a null frame

0 = Received frame is a null frame

PPIS Payload Preamble Indicator Status (**vRF!Header!PPIndicator**)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

1 = Static segment: Network management vector at the beginning of the payload

Dynamic segment: Message ID at the beginning of the payload

0 = The payload segment of the received frame does not contain a network management vector or a message ID

RESS Reserved Bit Status (**vRF!Header!Reserved**)

Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

■ Output Buffer Command Mask (OBCM)

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by OBCR.OBRS[6:0]. Mask bits **RDSS** and **RHSS** are copied to the register internal storage when a Message RAM transfer is requested by **OBCR.REQ**. When OBF Host and OBF Shadow are swapped, mask bits **RDSH** and **RHSH** are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer. The data transfer between Output Buffer and Message RAM is described in detail in Section "Data Transfer from Message RAM to Output Buffer" on P. 1377 .

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBCM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSH	RHSH
0xD710	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSS	RHSS
	W															
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RHSS Read Header Section Shadow

- 1 = Header section selected for transfer from Message RAM to Output Buffer
- 0 = Header section is not read

RDSS Read Data Section Shadow

- 1 = Data section selected for transfer from Message RAM to Output Buffer
- 0 = Data section is not read

RHSH Read Header Section Host

- 1 = Header section selected for transfer from Message RAM to Output Buffer
- 0 = Header section is not read

RDSH Read Data Section Host

- 1 = Data section selected for transfer from Message RAM to Output Buffer
- 0 = Data section is not read

Note: After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag **MBC** of the selected message buffer in the MBSC1/2/3/4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag **ND** of the selected message buffer in the NDAT1/2/3/4 registers is cleared.

MB91460 Series

■ Output Buffer Command Request (OBCR)

After setting bit **REQ** to '1' while **OBSYS** is '0', **OBSYS** is automatically set to '1', **OBRs[6:0]** is copied to the register internal storage, mask bits **OBCM.RDSS** and **OBCM.RHSS** are copied to register OBCM internal storage, and the transfer of the message buffer selected by **OBRs[6:0]** from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to '0'.

By setting bit **VIEW** to '1' while **OBSYS** is '0', OBF Host and OBF Shadow are swapped. Additionally mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped with the register OBCM internal storage to keep them attached to the respective Output Buffer transfer. **OBRH[6:0]** signals the number of the message buffer currently accessible by the Host.

If bits **REQ** and **VIEW** are set to '1' with the same write access while **OBSYS** is '0', **OBSYS** is automatically set to '1' and OBF Shadow and OBF Host are swapped. Additionally mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards **OBRs[6:0]** is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to '0'.

Any write access to **OBCR[15:8]** while **OBSYS** is set will cause the error flag **EIR.IOBA** to be set. In this case the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in Section "Data Transfer from Message RAM to Output Buffer" on P. 1377 .

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBCR	R	0	0	0	0	0	0	0	0	0	OBRH6	OBRH5	OBRH4	OBRH3	OBRH2	OBRH1	OBRH0
0xD714	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	OBSYS	0	0	0	0	0		REQ	VIEW	0	OBRs6	OBRs5	OBRs4	OBRs3	OBRs2	OBRs1	OBRs0
	W																	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

OBRs[6:0] Output Buffer Request Shadow

Number of source message buffer to be transferred from the Message RAM to OBF Shadow. Valid values are 0x00 to 0x7F (0...127). If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see Section 58.4.10 FIFO Function (Page No.1370)) to OBF Shadow.

VIEW View Shadow Buffer

Toggles between OBF Shadow and OBF Host. Only writeable while **OBSYS** = '0'.

1 = Swap OBF Shadow and OBF Host

0 = No action

REQ Request Message RAM Transfer

Requests transfer of message buffer addressed by **OBRs[6:0]** from Message RAM to OBF Shadow. Only writeable while **OBSYS** = '0'.

1 = Transfer to OBF Shadow requested

0 = No request

OBSYS Output Buffer Busy Shadow

Set to '1' after setting bit **REQ**. When the transfer between the Message RAM and OBF Shadow has completed, **OBSYS** is set back to '0'.

1 = Transfer between Message RAM and OBF Shadow in progress

0 = No transfer in progress

OBRH[6:0] Output Buffer Request Host

Number of message buffer currently accessible by the Host via RDHS[1...3], MBS, and RDDS[1...64]. By writing **VIEW** to '1' OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host. Valid values are 0x00 to 0x7F (0...127).

58.4. Functional Description

This chapter describes the E-Ray implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification v2.1.

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

58.4.1 Communication Cycle

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

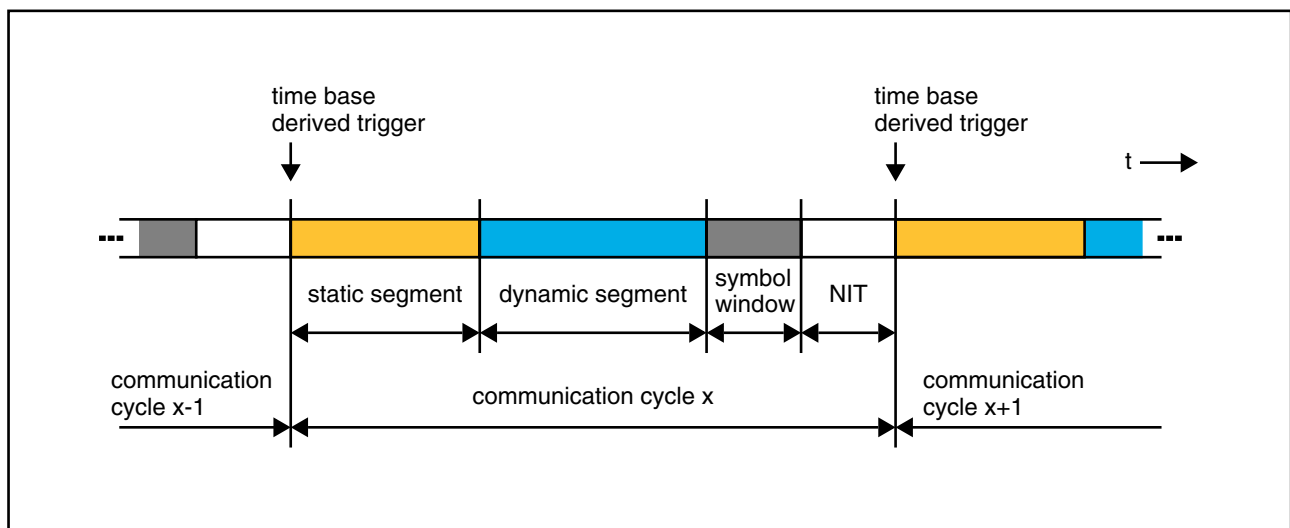


Figure 58.4-1 Structure of communication cycle

■ Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters: Number of Static Slots **GTUC7.NSS[9:0]**, Static Slot Length **GTUC7.SSL[9:0]**, Payload Length Static **MHDC.SFDL[6:0]**, Action Point Offset **GTUC9.APO[5:0]**

■ Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters: Number of Minislots **GTUC8.NMS[12:0]**, Minislot Length **GTUC8.MSL[5:0]**,
Minislot Action Point Offset **GTUC9.MAPO[4:0]**,
Start of Latest Transmit (last minislot) **MHDC.SLT[12:0]**

■ Symbol Window

During the symbol window only **one** media access test symbol (MTS) may be transmitted per channel. MTS symbols are send in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters: Symbol Window Action Point Offset **GTUC9.APO[4:0]** (same as for static slots),
Network Idle Time Start **GTUC4.NIT[13:0]**

■ Network Idle Time (NIT)

During network idle time the CC has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters: Network Idle Time Start **GTUC4.NIT[13:0]**,
Offset Correction Start **GTUC4.OCS[13:0]**

■ Configuration of NIT Start and Offset Correction Start

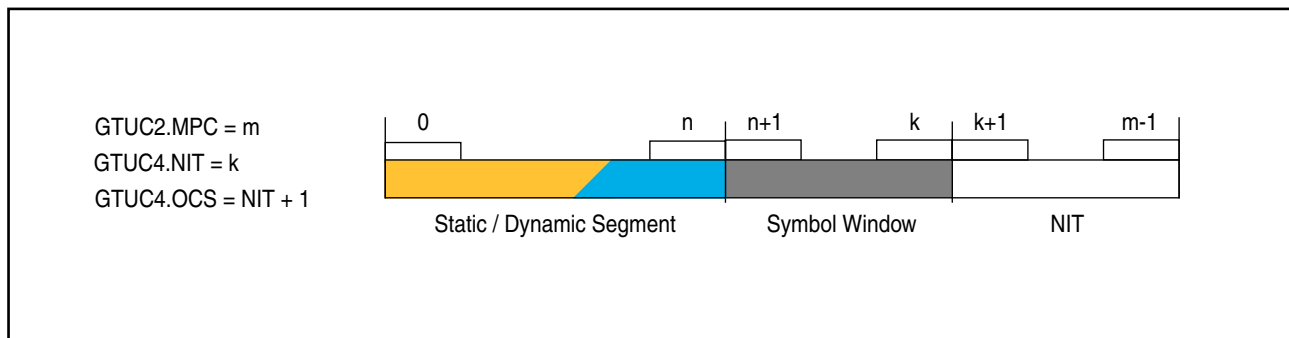


Figure 58.4-2 Configuration of NIT start and offset correction start

The number of macroticks per cycle **gMacroPerCycle** is assumed to be **m**. It is configured by programming **GTUC2.MPC** = **m**.

The static / dynamic segment starts with macrotick 0 and ends with macrotick **n**:

$n = \text{static segment length} + \text{dynamic segment offset} + \text{dynamic segment length} - 1\text{MT}$

$n = \text{gNumberOfStaticSlots} \cdot \text{gdStaticSlot} + \text{dynamic segment offset}$
 $+ \text{gNumberOfMinislots} \cdot \text{gdMinislot} - 1\text{MT}$

The static segment length is configured by **GTUC7.SSL** and **GTUC7.NSS**.

The dynamic segment length is configured by **GTUC8.MSL** and **GTUC8.NMS**.

The dynamic segment offset is:

If **gdActionPointOffset** ≤ **gdMinislotActionPointOffset**:

dynamic segment offset = 0 MT

Else if **gdActionPointOffset** > **gdMinislotActionPointOffset**:

dynamic segment offset = **gdActionPointOffset** - **gdMinislotActionPointOffset**

The NIT starts with macrotick **k+1** and ends with the last macrotick of cycle **m-1**. It has to be configured by setting **GTUC4.NIT** = **k**.

For the E-Ray the offset correction start is required to be **GTUC4.OCS** ≥ **GTUC4.NIT** + 1 = **k+1**.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by **k - n**.

58.4.2 Communication Modes

The FlexRay Protocol Specification v2.1 defines the Time-Triggered Distributed (TT-D) mode.

■ Time-triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- **Pure static:**Minimum 2 static slots + symbol window (optional)
- **Mixed static/dynamic:**Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

MB91460 Series

58.4.3 Clock Synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

■ Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time, even though each individual node maintains its own view of it. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (μT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

■ Local Time

Internally, nodes time their behaviour with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (μT).

Node specific:

- Oscillator clock -> prescaler -> microtick (μT)
- μT = basic unit of time measurement in a CC, clock correction is done in units of μTs
- Cycle counter + macrotick counter = nodes local view of the global time

■ Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (**GTUC2.SNM[3:0]**) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see FlexRay protocol specification v2.1, chapter 8.

● Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of **every** communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Correction done in **odd** numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened / shortened)

● Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of **odd** numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of μ Ts
- Distributed over macroticks comprising the next **even / odd** cycle pair (MTs lengthened / shortened)

● Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit **MRC.SPLM** has to be programmed to '1'.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames **SUCC1.TXSY** must be set to '1'.

■ External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is **not** checked against configured limits

MB91460 Series

58.4.4 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set **EIR.PEMC** and may trigger an interrupt to the Host if enabled. The actual error mode is signalled by **CCEV.ERRM[1:0]**.

Error Mode	Activity
ACTIVE (green)	Full operation , State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers EIR and SIR.
PASSIVE (yellow)	Reduced operation , State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers EIR and SIR.
COMM_HALT (red)	Operation halted , State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers EIR and SIR. The bus drivers are disabled.

Table 58.4-1 Error modes of the POC (degradation model)

■ Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the "maximum without clock correction passive" limit defined by **SUCC3.WCP[3:0]**, the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the "maximum without clock correction fatal" limit defined by **SUCC3.WCF[3:0]**, it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter **CCEV.CCFC[3:0]** allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any **odd** communication cycle during which either the missing offset correction **SFS.MOCS** or the missing rate correction **SFS.MRCS** flag is set.

The Clock Correction Failed Counter is reset to zero at the end of an **odd** communication cycle if neither the missing offset correction **SFS.MOCS** nor the missing rate correction **SFS.MRCS** flag is set.

The Clock Correction Failed Counter stops incrementing when the "maximum without clock correction fatal" value **SUCC3.WCF[3:0]** is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

Note: The transition to HALT state is prevented if **SUCC1.HCSE** is not set.

■ Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. **SUCC1.PTA[4:0]** defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If **SUCC1.PTA[4:0]** is set to zero the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

■ HALT Command

In case the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing **SUCC1.CMD[3:0] = "0110"**. In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from **CCSV.PSL[5:0]**.

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state **SUCC1.CMD[3:0]** will be reset to "0000" = command_not_accepted and bit **EIR.CNA** is set to '1'. If enabled an interrupt to the Host is generated.

■ FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing **SUCC1.CMD[3:0] = "0111"**. The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from **CCSV.PSL[5:0]**.

58.4.5 Communication Controller States

■ Communication Controller State Diagram

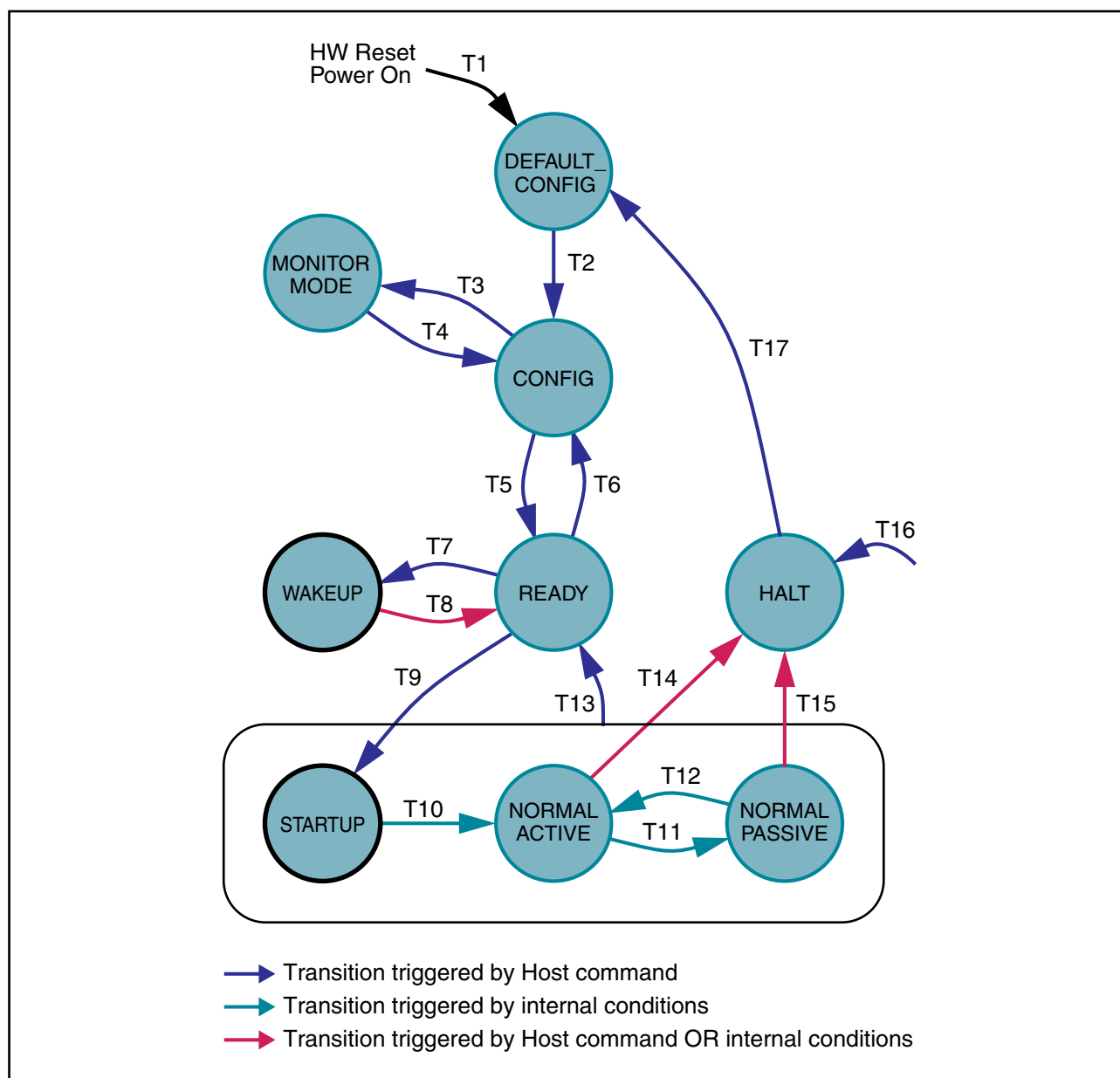


Figure 58.4-3 Overall state diagram of E-Ray communication controller

State transitions are controlled by external pins **eray_reset** and **eray_rxd1,2**, by the POC state machine, and by the CHI Command Vector **SUCC1.CMD[3:0]**.

The CC exits from **all** states to HALT state after application of the FREEZE command. (**SUCC1.CMD[3:0]** = "0111").

T#	Condition	From	To
1	Hard reset	All States	DEFAULT_CONFIG
2	Command CONFIG, SUCC1.CMD[3:0] = "0001"	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command MONITOR_MODE, SUCC1.CMD[3:0] = "1011"	CONFIG	MONITOR_MODE
4	Command CONFIG, SUCC1.CMD[3:0] = "0001"	MONITOR_MODE	CONFIG
5	Unlock sequence followed by command READY, SUCC1.CMD[3:0] = "0010"	CONFIG	READY
6	Command CONFIG, SUCC1.CMD[3:0] = "0001"	READY	CONFIG
7	Command WAKEUP, SUCC1.CMD[3:0] = "0011"	READY	WAKEUP
8	Complete, non-aborted transmission of wakeup pattern OR received WUP OR received frame header OR wakeup collision OR command READY, SUCC1.CMD[3:0] = "0010"	WAKEUP	READY
9	Command RUN, SUCC1.CMD[3:0] = "0100"	READY	STARTUP
10	Successful startup	STARTUP	NORMAL_ACTIVE
11	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by SUCC3.WCP[3:0]	NORMAL_ACTIVE	NORMAL_PASSIVE
12	Number of valid correction terms reached the Passive to Active limit configured by SUCC1.PTA[4:0]	NORMAL_PASSIVE	NORMAL_ACTIVE
13	Command READY, SUCC1.CMD[3:0] = "0010"	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
14	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by SUCC3.WCF[3:0] AND bit SUCC1.HCSE set to '1' OR command HALT, SUCC1.CMD[3:0] = "0110"	NORMAL_ACTIVE	HALT
15	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by SUCC3.WCF[3:0] AND bit SUCC1.HCSE set to '1' OR command HALT, SUCC1.CMD[3:0] = "0110"	NORMAL_PASSIVE	HALT
16	Command FREEZE, SUCC1.CMD[3:0] = "0111"	All States	HALT
17	Command CONFIG, SUCC1.CMD[3:0] = "0001"	HALT	DEFAULT_CONFIG

Table 58.4-2 State transitions of E-Ray overall state machine

MB91460 Series

■ DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the physical layer is in inactive state.

The CC enters this state

- When leaving hard reset (external reset signal **eray_reset** is deactivated)
- When exiting from HALT state

To leave DEFAULT_CONFIG state the Host has to write **SUCC1.CMD[3:0]** = "0001". The CC then transits to CONFIG state.

■ CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from MONITOR_MODE or READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyse status information and configuration. Before leaving CONFIG state the Host has to assure that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described in "[Lock Register \(LCK\)](#)" on P. 1264 . Directly after unlocking the CONFIG state the Host has to write **SUCC1.CMD[3:0]** to enter the next state.

Note: Status bits MHDS[14:0], registers TXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (**ERAY_SCLK**, **CLKB**). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

■ MONITOR_MODE

After unlocking CONFIG state and writing **SUCC1.CMD[3:0]** = "1011" the CC enters MONITOR_MODE. In this mode the CC is able to receive FlexRay frames and to detect wakeup pattern. The temporal integrity of received frames is not checked, and therefore cycle counter filtering is not supported. This mode can be used for debugging purposes in case e.g. that startup of a FlexRay network fails. After writing **SUCC1.CMD[3:0]** = "0001" the CC transits back to CONFIG state.

In MONITOR_MODE the pick first valid mechanism is disabled. This means that a receive message buffer may only be configured to receive on one channel. Received frames are stored into message buffers according to frame ID and receive channel. Null frames are handled like data frames. After frame reception only status bits **MBS.VFRA**, **MBS.VFRB**, **MBS.MLST**, **MBS.RCIS**, **MBS.SFIS**, **MBS.SYNS**, **MBS.NFIS**, **MBS.PPIS**, **MBS.RESS** have valid values.

In MONITOR_MODE the CC is not able to distinguish between CAS and MTS symbols. In case one of these symbols is received on one or both of the two channels, the flags **SIR.MTSA** resp. **SIR.MTSB** are set. **SIR.CAS** has no function in MONITOR_MODE.

■ READY State

After unlocking CONFIG state and writing **SUCC1.CMD[3:0]** = "0010" the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing **SUCC1.CMD[3:0]** = "0010" (READY command).

The CC exits from this state

- To CONFIG state by writing **SUCC1.CMD[3:0]** = "0001" (CONFIG command)
- To WAKEUP state by writing **SUCC1.CMD[3:0]** = "0011" (WAKEUP command)
- To STARTUP state by writing **SUCC1.CMD[3:0]** = "0100" (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

Note: Status bits MHDS[14:0], registers TXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

MB91460 Series

■ WAKEUP State

The description below is intended to help configuring wakeup for the E-Ray IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

- When exiting from READY state by writing **SUCC1.CMD[3:0]** = "0011" (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing **SUCC1.CMD[3:0]** = "0010" (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an **external** wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing **SUCC1.WUCS**. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag **SIR.WST**. The wakeup status vector can be read from **CCSV.WSV[2:0]**. If a valid wakeup pattern was received also either flag **SIR.WUPA** or flag **SIR.WUPB** is set.

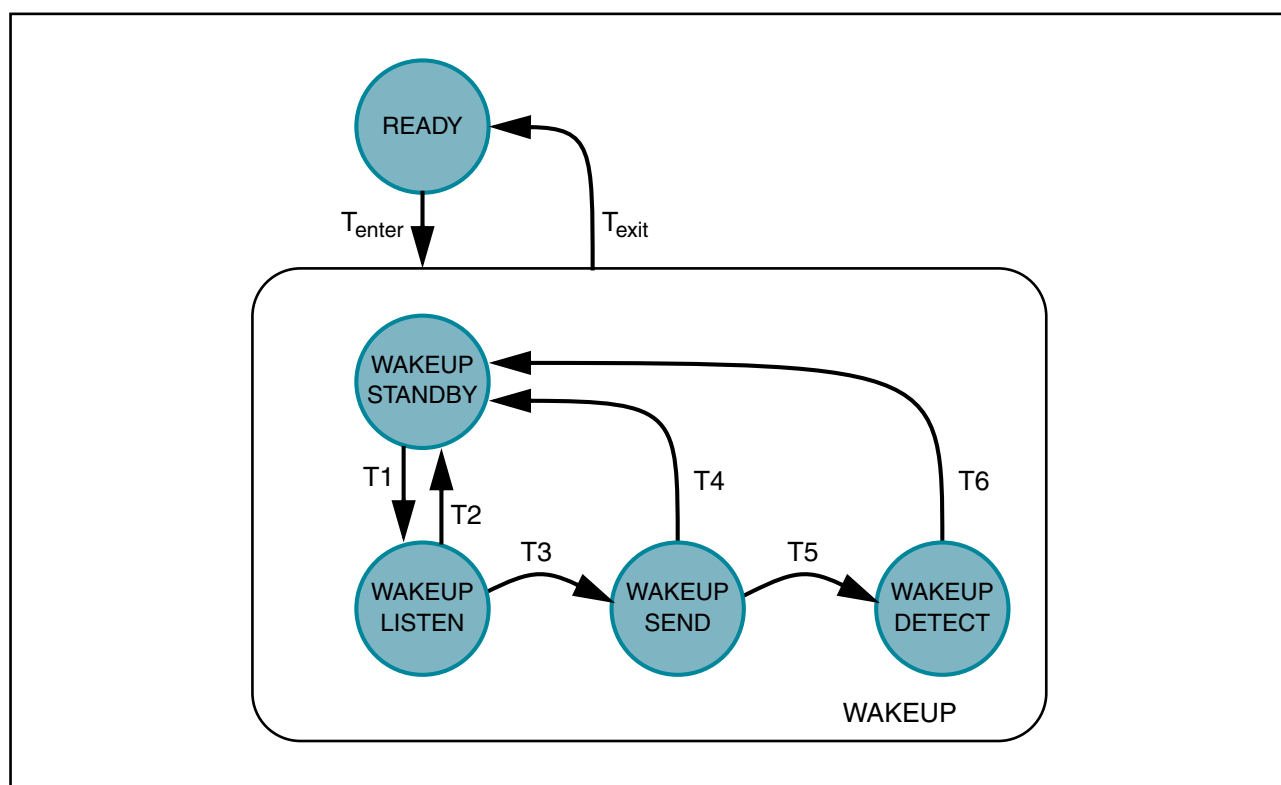


Figure 58.4-4 Structure of POC state WAKEUP

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing SUCC1.CMD[3:0] = "0011" (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit SUCC1.WUCS OR frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired OR WUP detected on wakeup channel selected by bit SUCC1.WUCS OR frame header received on either available channel	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) OR Host commands change to READY state by writing SUCC1.CMD[3:0] = "0010" (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

Table 58.4-3 State transitions WAKEUP

MB91460 Series

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters listen timeout **SUCC2.LT[20:0]** and listen timeout noise **SUCC2.LTN[3:0]**. Listen timeout enables a fast cluster wakeup in case of a noise free environment, while listen timeout noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by **SUCC2.LT[20:0]**. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification v2.1 recommends that two different CCs shall awake the two channels.

● Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signalled to their local Host.

- Wakeup procedure controlled by Host (single-channel wakeup):
- Configure the CC in CONFIG state
 - Select wakeup channel by programming bit **SUCC1.WUCS**
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing **SUCC1.CMD[3:0] = "0011"**
 - CC enters WAKEUP
 - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
 - In a dual channel cluster wait for WUP on the other channel
 - Reset coldstart inhibit flag **CCSV.CSI** by writing **SUCC1.CMD[3:0] = "1001"** (ALLOW_COLDSTART command)
- Command CC to enter startup by writing **SUCC1.CMD[3:0] = "0100"** (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing **SUCC1.CMD[3:0] = "0100"** (RUN command)

● Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers PRTC1 and PRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by **PRTC2.TXL[5:0]**
- Wakeup symbol idle time used to listen for activity on the bus, configured by **PRTC2.TXI[7:0]**
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by **PRTC1.RWP[5:0]** (2 to 63 repetitions)
- Wakeup symbol receive window length configured by **PRTC1.RXW[8:0]**
- Wakeup symbol receive low time configured by **PRTC2.RXL[5:0]**
- Wakeup symbol receive idle time configured by **PRTC2.RXI[5:0]**

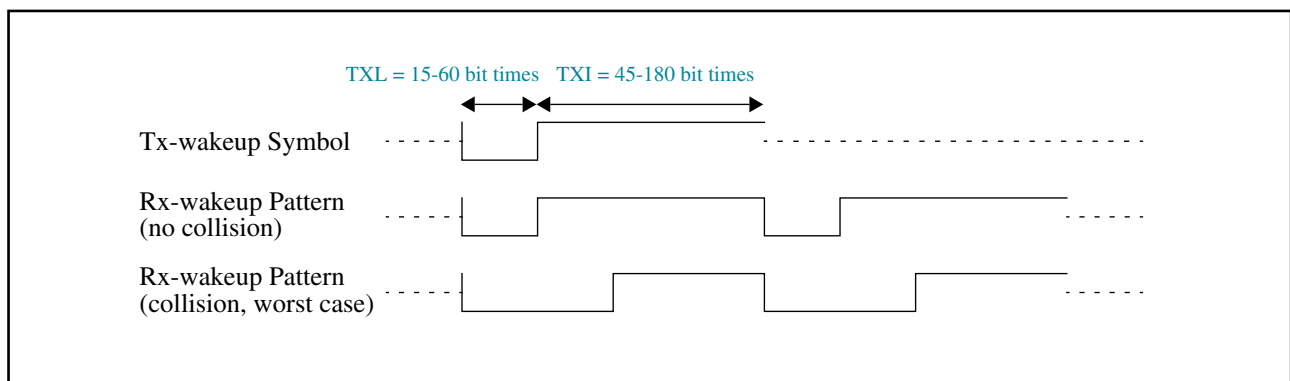


Figure 58.4-5 Timing of wakeup pattern

MB91460 Series

■ STARTUP State

The description below is intended to help configuring startup for the E-Ray IP-module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.2.

Any node entering STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via (see figure 58.4-6):

- Coldstart path initiating the schedule synchronization (leading coldstart node)
- Coldstart path joining other coldstart nodes (following coldstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits **SUCC1.TXST** and **SUCC1.TXSY** set to '1'. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by **SUCC1.CSA[4:0]**.

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

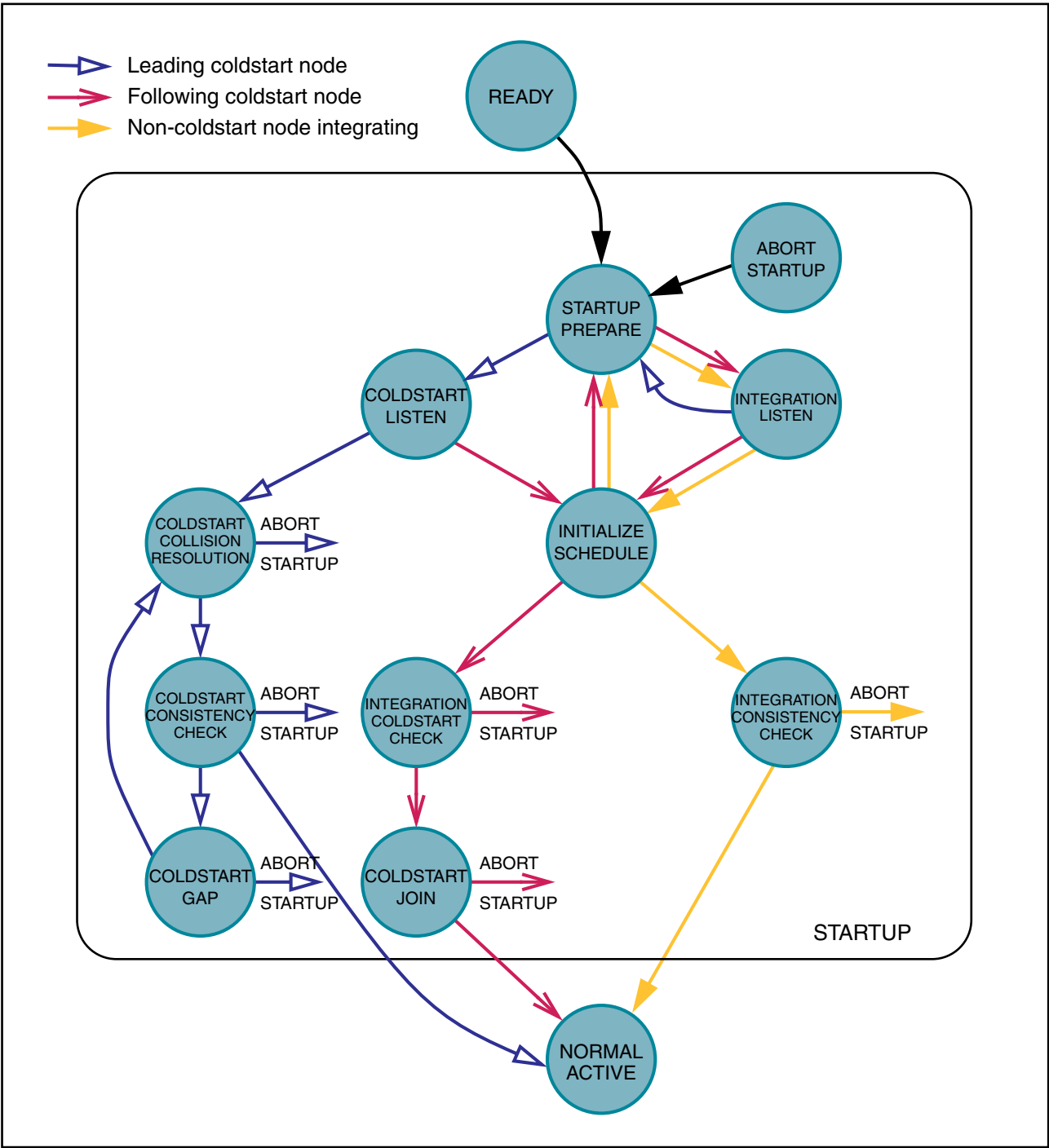


Figure 58.4-6 State diagram time-triggered startup

MB91460 Series

● Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit **CCSV.CSI** is set, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit **CCSV.CSI** is set whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (**SUCC1.CMD[3:0]** = "1001")

● Startup Timeouts

The CC supplies two different μ T timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

Note: The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values **SUCC2.LT[20:0]** and **SUCC2.LTN[3:0]**.

Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming **SUCC2.LT[20:0]**(see "SUC Configuration Register 2 (SUCC2)" on P. 1286).

The startup timeout is: **pdListenTimeout** = **SUCC2.LT[20:0]**

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming **SUCC2.LTN[3:0]** (see "SUC Configuration Register 2 (SUCC2)" on P. 1286).

The startup noise timeout is:

$$\text{pdListenTimeout} \cdot \text{gListenNoise} = \text{SUCC2.LT}[20:0] \cdot (\text{SUCC2.LTN}[3:0] + 1)$$

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

● Path of leading Coldstart Node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by **SUCC1.CSA[4:0]**. The number of remaining coldstarts attempts can be read from **CCSV.RCA[4:0]**. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

● Path of following Coldstart Node (responding to leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

● Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signalled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

■ NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even / odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing **SUCC1.CMD[3:0] = "0110"**
(HALT command, at the end of the current cycle)
- HALT state by writing **SUCC1.CMD[3:0] = "0111"** (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing **SUCC1.CMD[3:0] = "0010"** (READY command)

■ NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing **SUCC1.CMD[3:0] = "0110"**
(HALT command, at the end of the current cycle)
- HALT state by writing **SUCC1.CMD[3:0] = "0111"** (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE.
The transition takes place when **CCEV.PTAC[4:0]** equals **SUCC1.PTA[4:0] - 1**
- To READY state by writing **SUCC1.CMD[3:0] = "0010"** (READY command)

■ HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing **SUCC1.CMD[3:0]** = "0110" (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing **SUCC1.CMD[3:0]** = "0111" (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and **SUCC1.HCSE** is set
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and **SUCC1.HCSE** is set

The CC exits from this state to DEFAULT_CONFIG state

- By writing **SUCC1.CMD[3:0]** = "0001" (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analysing purposes.

When the Host writes **SUCC1.CMD[3:0]** = "0110" (HALT command), the CC sets bit **CCSV.HRQ** and enters HALT state at the next end of cycle.

When the Host writes **SUCC1.CMD[3:0]** = "0111" (FREEZE command), the CC enters HALT state immediately and sets bit **CCSV.FSI**.

The POC state from which the transition to HALT state took place can be read from **CCSV.PSL[5:0]**.

58.4.6 Network Management

The accrued Network Management (NM) vector can be read from registers NMV1...3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (**PPI**) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by **NEMC.NML[3:0]**. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the **PPI** bit set, bit **PPIT** in the header section of the respective transmit buffer has to be set via **WRHS1.PPIT**. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

Note: In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by **NEMC.NML[3:0]**.

When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case NMV1...3 holds the value from the cycle before.

MB91460 Series

58.4.7 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance / transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

Note: For the FIFO the acceptance filter is configured by the FIFO Rejection Filter and the FIFO Rejection Filter Mask.

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

■ Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the **lowest** message buffer number is used.

■ Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 resp. 1 is configured to hold the startup / sync frame or the single slot frame by bits **SUCC1.TXST**, **SUCC1.TXSY**, and **SUCC1.TSM**, cycle counter filtering for message buffer 0 resp. 1 must be disabled.

Note: Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is **not** allowed.

The set of cycle numbers belonging to a cycle set is determined as described in table 58.4-4.

Cycle Code	Matching Cycle Counter Values		
0b000000x	all Cycles		
0b000001c	every second Cycle	at (Cycle Count)mod2	= c
0b00001cc	every fourth Cycle	at (Cycle Count)mod4	= cc
0b0001ccc	every eighth Cycle	at (Cycle Count)mod8	= ccc
0b001cccc	every sixteenth Cycle	at (Cycle Count)mod16	= cccc
0b01ccccc	every thirty-second Cycle	at (Cycle Count)mod32	= ccccc
0b1ccccc	every sixty-fourth Cycle	at (Cycle Count)mod64	= cccccc

Table 58.4-4 Definition of cycle set

Table 58.4-5 below gives some examples for valid cycle sets to be used for cycle counter filtering:

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7- -63 ↴
0b0000100	0-4-8-12- -60 ↴
0b0001110	6-14-22-30- -62 ↴
0b0011000	8-24-40-56 ↴
0b0100011	3-35 ↴
0b1001001	9 ↴

Table 58.4-5 Examples for valid cycle sets

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Other filter criteria must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Other filter criteria must also be met.

MB91460 Series

■ Channel ID Filtering

There is a 2-bit channel filtering field (**CHA**, **CHB**) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see table 58.4-6).

CHA	CHB	Transmit Buffer transmit frame	Receive Buffer store valid receive frame
1	1	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)
1	0	on channel A	received on channel A
0	1	on channel B	received on channel B
0	0	no transmission	ignore frame

Table 58.4-6 Channel filtering configuration

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (**CHA** and **CHB** set).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (**CHA** and **CHB** set).

Note: If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no frames are transmitted resp. received frames are ignored (same function as **CHA = CHB = '0'**).

■ FIFO Filtering

For FIFO filtering there is one rejection filter and one rejection filter mask available. The FIFO filter consists of channel filter **FRF.CH[1:0]**, frame ID filter **FRF.FID[10:0]**, and cycle counter filter **FRF.CYF[6:0]**. Registers FRF and FRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles **not** belonging to the cycle set specified by **FRF.CYF[6:0]**, **all** frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

58.4.8 Transmit Process

■ Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

■ Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by **MHDC.SLT[12:0]** defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

■ Transmit Buffers

E-Ray message buffers can be configured as transmit buffers by programming bit **CFG** in the header section of the respective message buffer to '1' via WRHS1.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment:channel A **or** channel B,
channel A **and** channel B
- Dynamic segment:channel A **or** channel B

Message buffer 0 resp. 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by **SUCC1.TXST**, **SUCC1.TXSY**, and **SUCC1.TSM**. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of **MRC.SEC[1:0]** (see ["Reconfiguration of Message Buffers" on P. 1372](#)). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the **PPIT** bit in the header section of the respective message buffer to '1' and write the network management information to the data section of the message buffer (see ["Network Management" on P. 1362](#)).

MB91460 Series

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by **MHDC.SFDL[6:0]**, the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is logical zero.

Note: In case of an odd payload length (PLC = 1,3,5,...) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is all zero.

Each transmit buffer provides a transmission mode flag **TXM** that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In **single-shot mode** the CC resets the respective **TXR** flag after transmission has completed. Now the Host may update the transmit buffer.

In **continuous mode**, the CC does not reset the respective transmission request flag **TXR** after successful transmission. In this case a frame is sent out each time the filter criteria match. The **TXR** flag can be reset by the Host by writing the respective message buffer number to the IBCR register while bit **IBCM.STXRH** is set to '0'.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

■ Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via WRHS1, WRHS2, and WRHS3
- Write the data section of the transmit buffer via WRDSn
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register IBCR
- If configured in register IBCM, the transmission request flag **TXR** for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective **TXR** bit (**TXR** = '0') in the TRXQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective **TXR** flag in the TXRQ1/2/3/4 register is reset (single-shot mode), and, if bit **MBI** in the header section of the message buffer is set, flag **SIR.TXI** is set to '1'. If enabled, an interrupt is generated.

■ Null Frame Transmission

If in static segment the Host does not set the transmission request flag before transmit time, and if there is no other transmit buffer with matching filter criteria, the CC transmits a null frame with the null frame indication bit **set to '0'** and the payload data **set to zero**.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (**TXR** = '0').
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status MBS is updated.

Null frames are not transmitted in the dynamic segment.

58.4.9 Receive Process

■ Dedicated Receive Buffers

A portion of the E-Ray message buffers can be configured as dedicated receive buffers by programming bit **CFG** in the header section of the respective message buffer to '0' via WRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A **or** channel B,
channel A **and** channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A **or** channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of **MRC.SEC[1:0]** (see ["Reconfiguration of Message Buffers" on P. 1372](#)). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

■ Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via WRHS1, WRHS2, and WRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register IBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective **ND** flag in the NDAT1/2/3/4 registers is set, and, if bit **MBI** in the header section of that message buffer is set, flag **SIR.RXI** is set to '1'. If enabled, an interrupt is generated.

In case that bit **ND** was already set when the Message Handler updates the message buffer, bit **MBS.MLST** of the respective message buffer is set and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status MBS is updated.

When the Message Handler changed the message buffer status MBS in the header section of a message buffer, the respective **MBC** flag in the MBSC1/2/3/4 registers is set, and if bit **MBI** in the header section of that message buffer is set, flag **SIR.MBSI** is set to '1'. If enabled an interrupt is generated.

MB91460 Series

If the payload length of a received frame **PLR[6:0]** is longer than the value programmed by **PLC[6:0]** in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in ["Data Transfer from Message RAM to Output Buffer" on P. 1377](#) .

Note: The **ND** and **MBC** flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

■ Null Frame Reception

The payload segment of a received null frame is **not** copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status MBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status MBS in the header section of a message buffer, the respective **MBC** flag in the MBSC1/2/3/4 register is set, and if bit **MBI** in the header section of that message buffer is set, flag **SIR.MBSI** is set to '1'. If enabled, an interrupt is generated.

58.4.10 FIFO Function

■ Description

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by **MRC.FFB[7:0]** and ending with the message buffer referenced by **MRC.LCB[7:0]**. Up to 127 message buffers can be assigned to the FIFO.

Every **valid** incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status MBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit **SIR.RFNE** shows that the FIFO is not empty, bit **SIR.RFCL** is set when the receive FIFO fill level **FSR.RFFL[7:0]** is equal or greater than the critical level as configured by **FCL.CL[7:0]**, bit **EIR.RFO** shows that a FIFO overrun has been detected. If enabled, interrupts are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag **EIR.RFO**.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag **SIR.RFNE** is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in figure 58.4-7 for a three message buffer FIFO.

The programmable FIFO Rejection Filter (FRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit **FRF.RSS** is set to '1' (default), all messages received in the static segment are rejected by the FIFO. If bit **FRF.RNF** is set to '1' (default), received null frames are not stored in the FIFO.

The FIFO Rejection Filter Mask (FRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

MB91460 Series

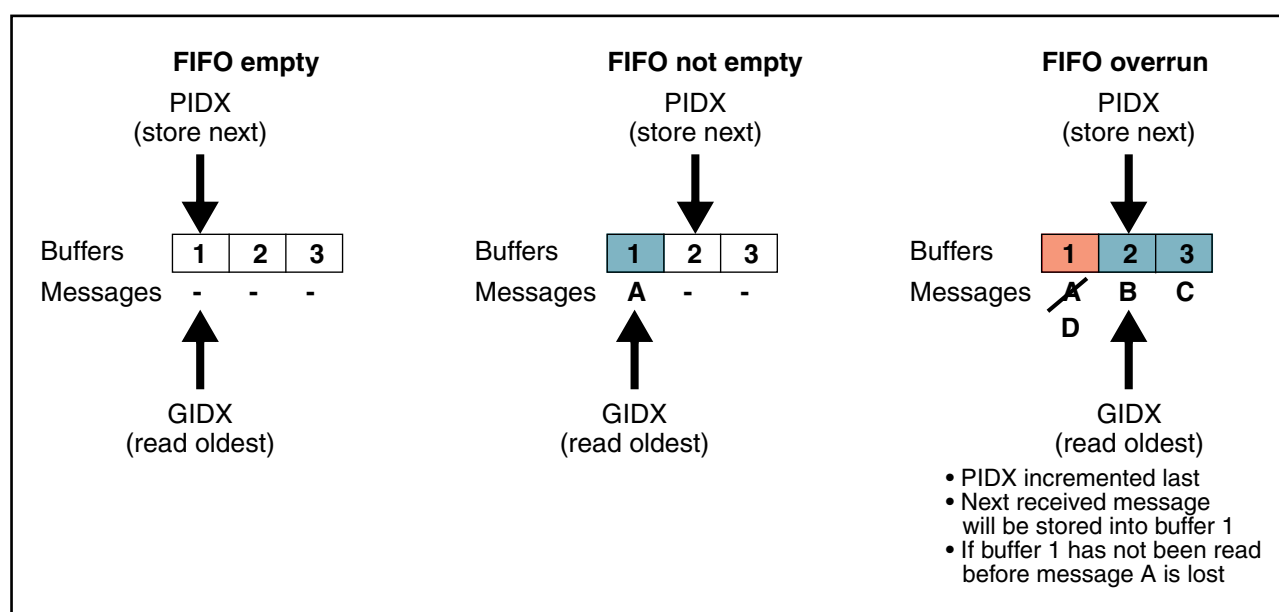


Figure 58.4-7 FIFO status: empty, not empty, overrun

■ Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via **WRHS2.PLC[6:0]**. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via **WRHS3.DP[10:0]**.

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

Note: It is recommended to program the MBI bits of the message buffers belonging to the FIFO to '0' via **WRHS1.MBI** to avoid generation of RX interrupts.

If the payload length of a received frame is longer than the value programmed by **WRHS2.PLC[6:0]** in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

■ Access to the FIFO

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by **MRC.FFB[7:0]**) to the register OBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

58.4.11 Message Handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two Transient Buffer RAMs. All accesses to the internal RAMs are 32+1 bit accesses. The additional bit is used for parity checking.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to **GTUC7.NSS[9:0]**. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from **GTUC7.NSS[9:0] + 1** to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

■ Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers WRHS1...3.

Reconfiguration has to be enabled via control bits **MRC.SEC[1:0]** in the Message RAM Configuration register. If a message buffer has not been transmitted / updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission / reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted / updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to [Table 58.4-7](#) below:

Start of Scan in Slot	Scan for Slots
1	2...15, 1 (next cycle)
8	16...23, 1 (next cycle)
16	24...31, 1 (next cycle)
24	32...39, 1 (next cycle)
...	...

Table 58.4-7 Scan of Message RAM

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by **MRC.FDB[7:0]**. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by **MRC.FDB[7:0]**.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the "Static Buffers", it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the "Static + Dynamic Buffers", it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

Note: Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted / updated from a received frame.

■ Host access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to IBCR or OBCR register.

The IBCM and OBCM registers can be used to write / read header and data section of the selected message buffer separately.

If bit **IBCM.STXR** is set to = '1', the transmission request flag **TXR** of the selected message buffer is automatically set after the message buffer has been updated. If bit **IBCM.STXR** is reset to '0', the transmission request flag **TXR** of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host), while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

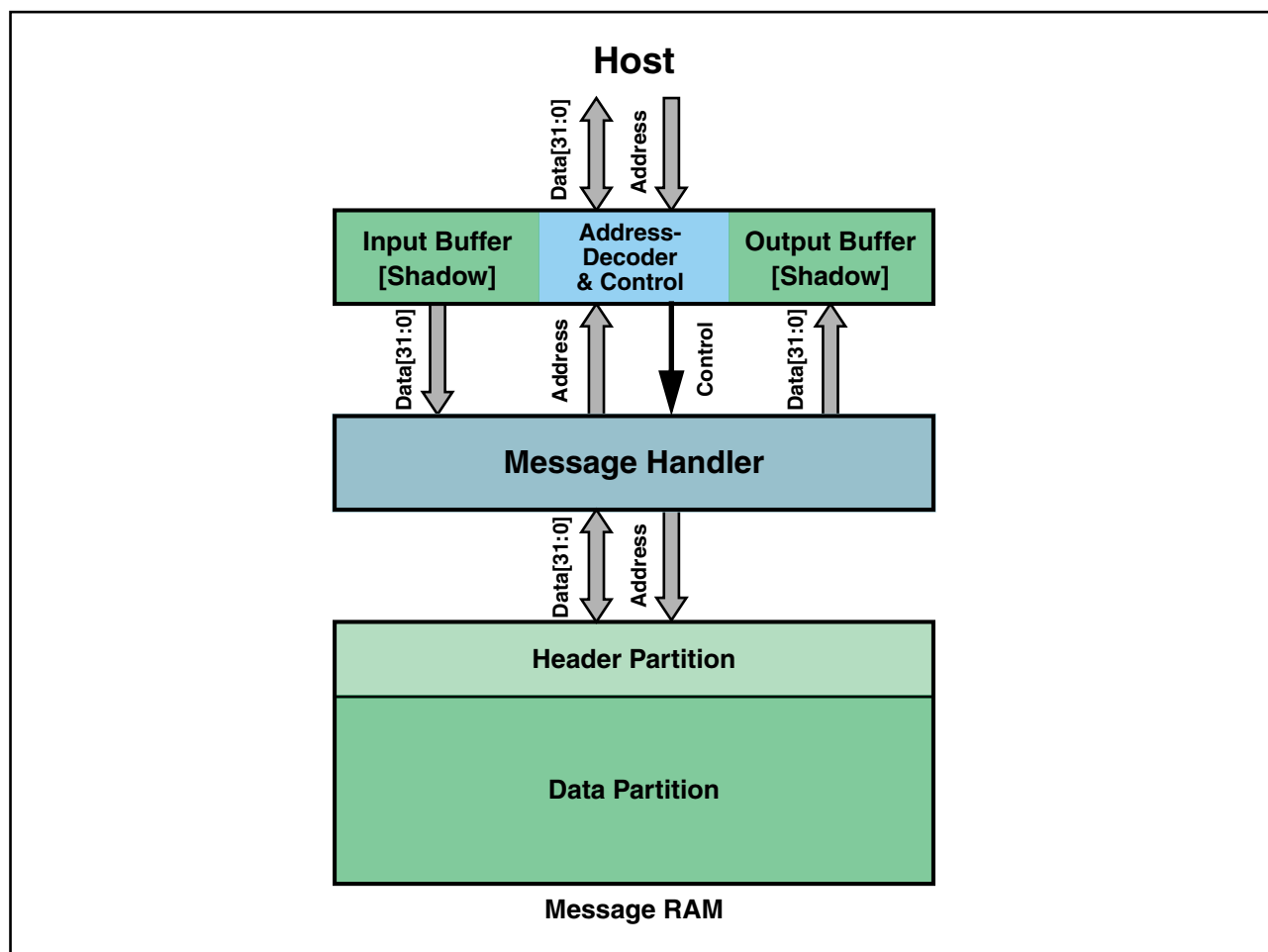


Figure 58.4-8 Host access to Message RAM

MB91460 Series

● Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to WRDSn and the header to WRHS1...3. The specific action is selected by configuring the Input Buffer Command Mask IBCM.

When the Host writes the number of the target message buffer in the Message RAM to **IBCR.IBRH[6:0]**, IBF Host and IBF Shadow are swapped (see figure 58.4-9).

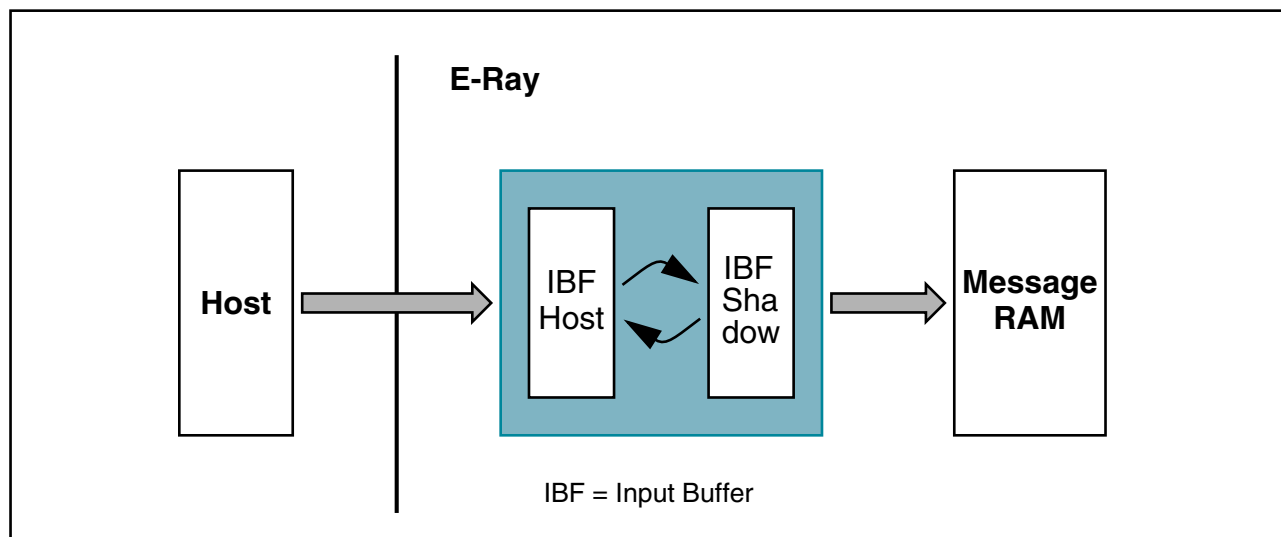


Figure 58.4-9 Double buffer structure Input Buffer

In addition the bits in the IBCM and IBCR registers are also swapped to keep them attached to the respective IBF section (see figure 58.4-10).

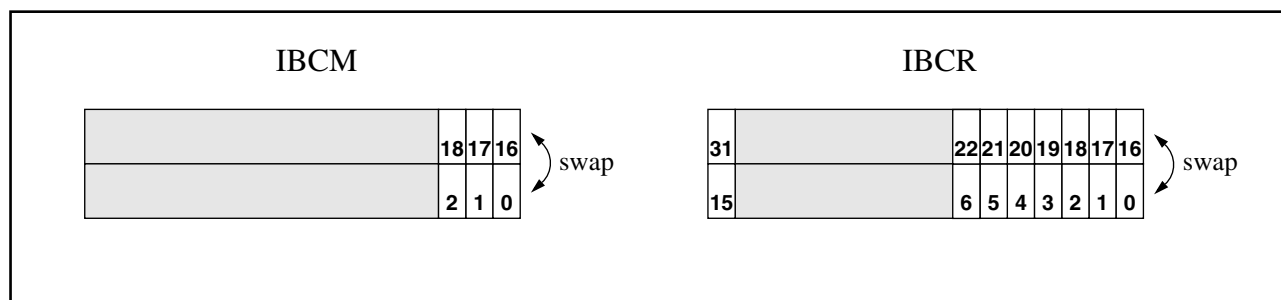


Figure 58.4-10 Swapping of IBCM and IBCR bits

With this write operation bit **IBCR.IBSYS** is set to '1'. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by **IBCR.IBRS[6:0]**.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit **IBCR.IBSYS** is set back to '0' and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to **IBCR.IBRH[6:0]**.

If a write access to **IBCR.IBRH[6:0]** occurs while **IBCR.IBSYS** is '1', **IBCR.IBSYH** is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, **IBCR.IBSYH** is reset to '0', **IBCR.IBSYS** remains set to '1', and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under **IBCR.IBRH[6:0]** and **IBCR.IBRS[6:0]** and the command mask flags are also swapped.

Example of a 8/16/32-bit Host access sequence:

Configure / update n-th message buffer via IBF

- Wait until **IBCR.IBSYH** is reset
- Write data section to WRDSn
- Write header section to WRHS1...3
- Write Command Mask: write **IBCM.STXRH**, **IBCM.LDSH**, **IBCM.LHSH**
- Demand data transfer to target message buffer: write **IBCR.IBRH[6:0]**

Configure / update (n+1)th message buffer via IBF

- Wait until **IBCR.IBSYH** is reset
- Write data section to WRDSn
- Write header section to WRHS1...3
- Write Command Mask: write **IBCM.STXRH**, **IBCM.LDSH**, **IBCM.LHSH**
- Demand data transfer to target message buffer: write **IBCR.IBRH[6:0]**

...

Note: Any write access to IBF while **IBCR.IBSYH** is '1' will set error flag **EIR.IIBA** to '1'. In this case the write access has no effect.

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 58.4-8 Assignment of IBCM bits

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22...16	r	IBRS[6:0]	IBF Request Shadow, number of message buffer currently / lately updated
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH[6:0]
6...0	r/w	IBRH[6:0]	IBF Request Host, number of message buffer to be updated next

Table 58.4-9 Assignment of IBCR bits

● Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register OBCR to trigger the data transfer as configured in OBCM. After the transfer has completed, the Host can read the transferred data from RDDSn, RDHS1...3, and MBS.

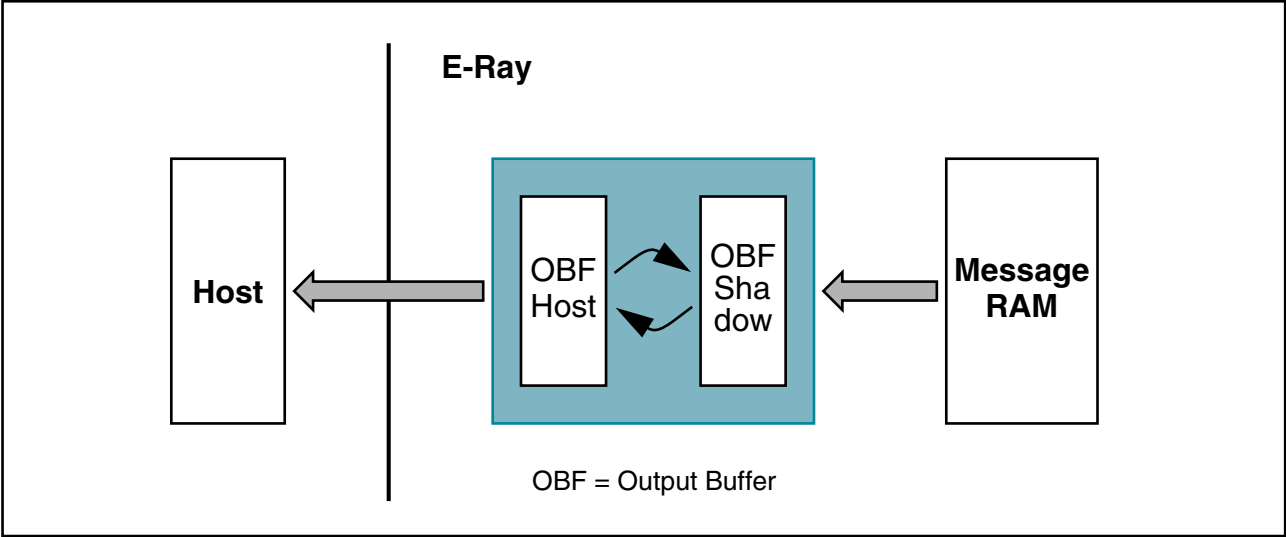


Figure 58.4-11 Double buffer structure Output Buffer

OBF Host and OBF Shadow as well as bits **OBCM.RHSS**, **OBCM.RDSS**, **OBCM.RHSH**, **OBCM.RDSH** and bits **OBCR.OBRS[6:0]**, **OBCR.OBRH[6:0]** are swapped under control of bits **OBCR.VIEW** and **OBCR.REQ**.

Writing bit **OBCR.REQ** to '1' copies bits **OBCM.RHSS**, **OBCM.RDSS** and bits **OBCR.OBRS[6:0]** to an internal storage (see figure 58.4-12).

After setting **OBCR.REQ** to '1', **OBCR.OBSYS** is set to '1', and the transfer of the message buffer selected by **OBCR.OBRS[6:0]** from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the **OBCR.OBSYS** bit is set back to '0'. Bits **OBCR.REQ** and **OBCR.VIEW** can only be set to '1' while **OBCR.OBSYS** is '0'.

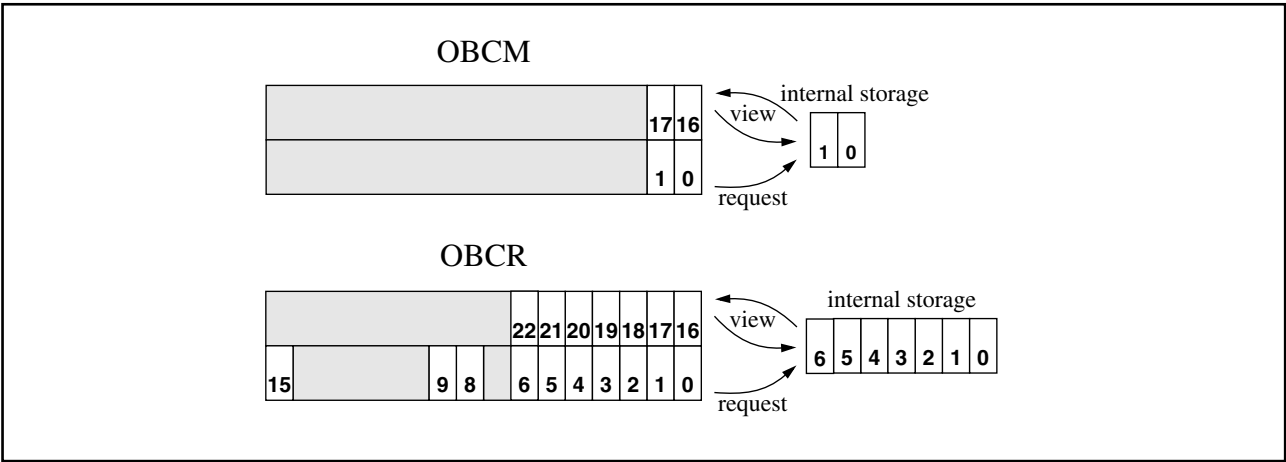


Figure 58.4-12 Swapping of OBCM and OBCR bits

OBF Host and OBF Shadow are swapped by setting bit **OBCR.VIEW** to '1' while bit **OBCR.OBSYS** is '0' (see figure 58.4-11).

In addition bits **OBCR.OBRH[6:0]** and bits **OBCM.RHSH**, **OBCM.RDSH** are swapped with the registers internal storage thus assuring that the message buffer number stored in **OBCR.OBRH[6:0]** and the mask configuration stored in **OBCM.RHSH**, **OBCM.RDSH** matches the transferred data stored in OBF Host (see figure 58.4-12).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits **REQ** and **VIEW** are set to '1' with the same write access while **OBSYS** is '0', **OBSYS** is automatically set to '1' and OBF Shadow and OBF Host are swapped. Additionally mask bits **OBCM.RDSH** and **OBCM.RHSH** are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards **OBRH[6:0]** is copied to the register internal storage, mask bits **OBCM.RDSS** and **OBCM.RHSS** are copied to register OBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signalled by setting **OBSYS** back to '0'.

Example of an 8/16/32-bit Host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to **OBCR.REQ** and **OBCR.VIEW** are necessary:

- Wait until **OBCR.OBSYS** is reset
- Write Output Buffer Command Mask **OBCM.RHSS**, **OBCM.RDSS**
- Request transfer of message buffer to OBF Shadow by writing **OBCR.OBRH[6:0]** and **OBCR.REQ** (in case of an 8-bit Host interface, **OBCR.OBRH[6:0]** has to be written **before OBCR.REQ**).
- Wait until **OBCR.OBSYS** is reset
- Toggle OBF Shadow and OBF Host by writing **OBCR.VIEW = '1'**
- Read out transferred message buffer by reading **RDDS_n**, **RDHS1...3**, and **MBS**

MB91460 Series

Example of an 8/16/32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until **OBCR.OBSYS** is reset
- Write Output Buffer Command Mask **OBCM.RHSS**, **OBCM.RDSS** for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing **OBCR.OBRS[6:0]** and **OBCR.REQ** (in case of an 8-bit Host interface, **OBCR.OBRS[6:0]** has to be written **before** **OBCR.REQ**).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until **OBCR.OBSYS** is reset
- Write Output Buffer Command Mask **OBCM.RHSS**, **OBCM.RDSS** for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing **OBCR.OBRS[6:0]** of 2nd message buffer, **OBCR.REQ**, and **OBCR.VIEW** (in case of an 8-bit Host interface, **OBCR.OBRS[6:0]** has to be written **before** **OBCR.REQ** and **OBCR.VIEW**).
- Read out 1st transferred message buffer by reading **RDDSn**, **RDHS1...3**, and **MBS**
- ...

Demand access to last requested message buffer without request of another message buffer:

- Wait until **OBCR.OBSYS** is reset
- Demand access to last transferred message buffer by writing **OBCR.VIEW**
- Read out last transferred message buffer by reading **RDDSn**, **RDHS1...3**, and **MBS**

Pos.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 58.4-10 Assignment of OBCM bits

Pos.	Access	Bit	Function
22...16	r	OBRH[6:0]	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6...0	r/w	OBRS[6:0]	OBF Request Shadow, number of message buffer for next request

Table 58.4-11 Assignment of OBCR bits

■ FlexRay Protocol Controller access to Message RAM

The two Transient Buffer RAMs (TBF A,B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Transient Buffer RAM is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be send to Transient Buffer Tx, the FlexRay Channel Protocol Controller can access Transient Buffer Rx to store the message it is actually receiving. During transmission of the message stored in Transient Buffer Tx, the Message Handler transfers the last received message stored in Transient Buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Transient Buffer RAMs and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

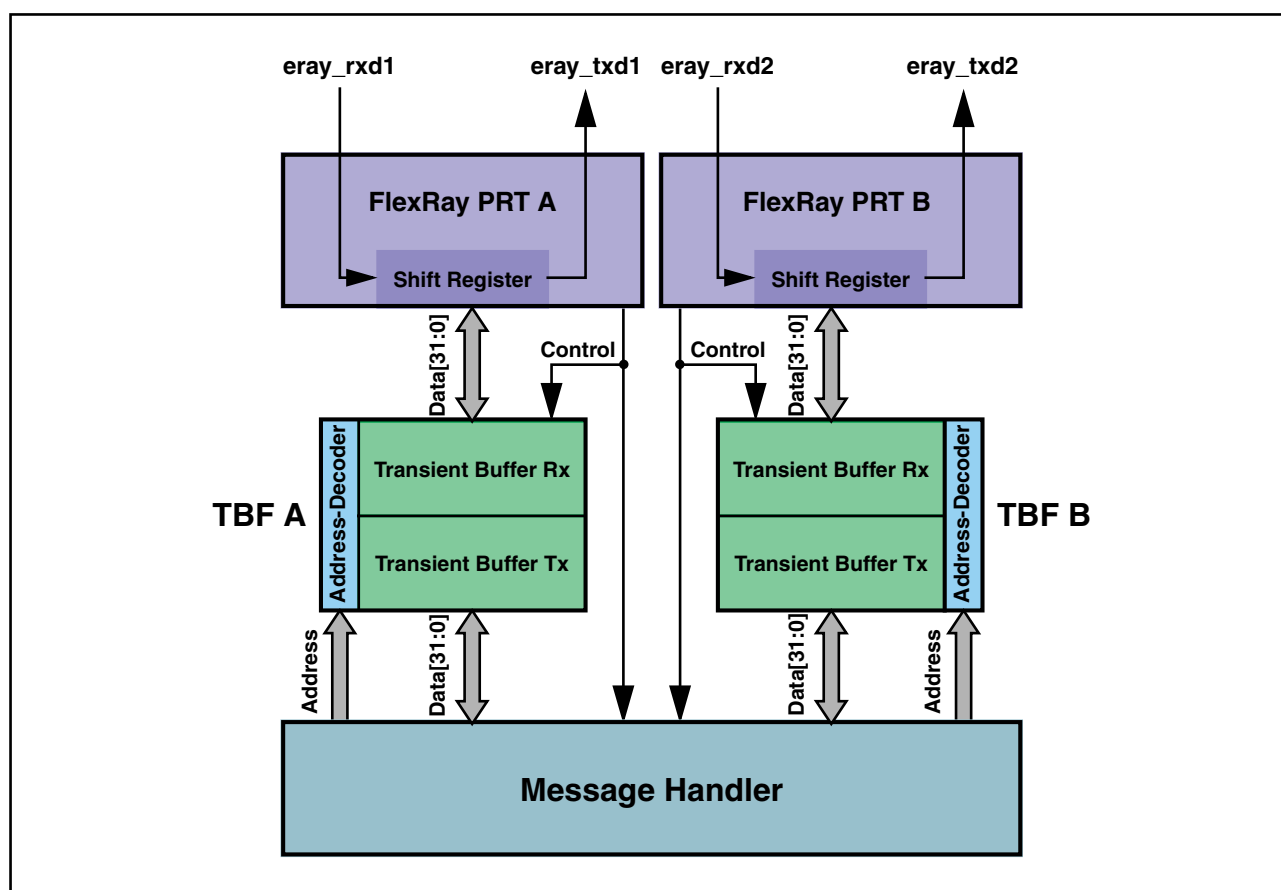


Figure 58.4-13 Access to Transient Buffer RAMs

MB91460 Series

58.4.12 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is organized $2048 \times 33 = 67,584$ bit. Each 32-bit word is protected by a parity bit. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0...254), the Message RAM has a structure as shown in figure 58.4-14.

The data partition is allowed to start at Message RAM word number: $(\text{MRC.LCB} + 1) \cdot 4$

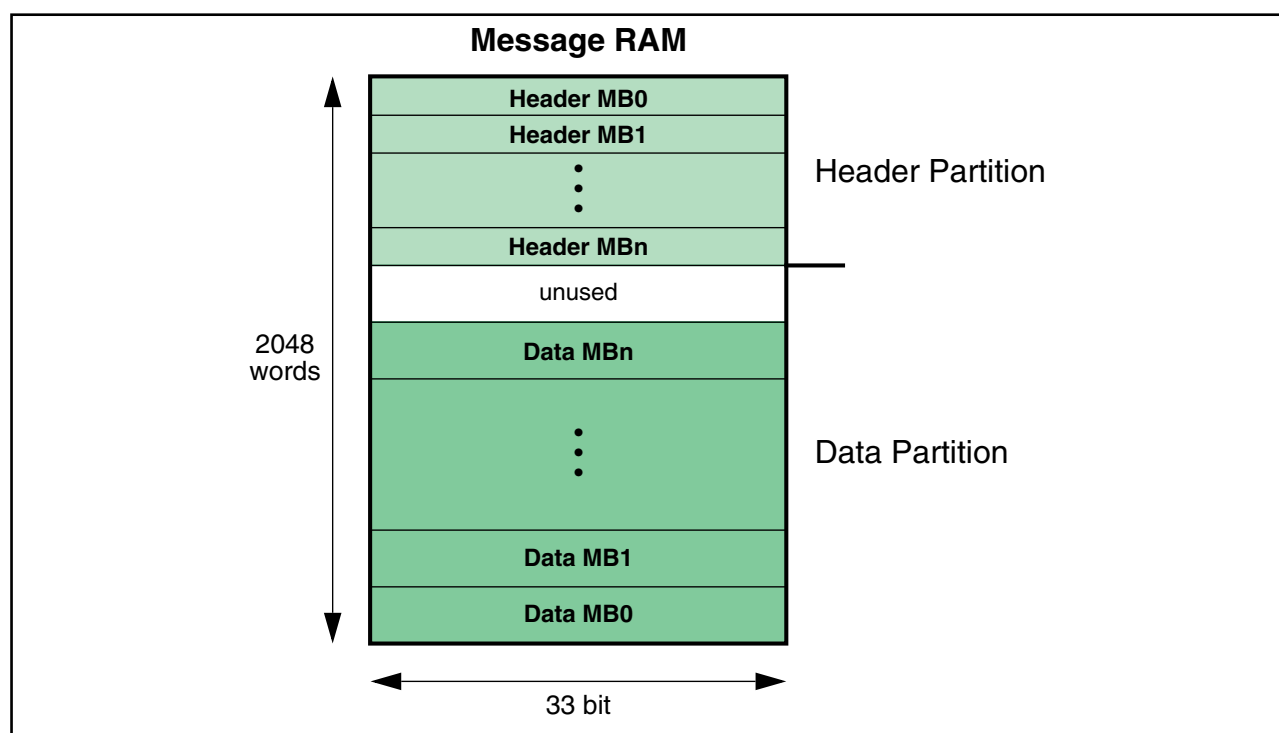


Figure 58.4-14 Configuration example of message buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32+1 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

Restriction: header partition + data partition may not occupy more than 2048 33-bit words.

■ Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in table 58.4-12 below. Configuration of the header sections of the message buffers is done via IBF (WRHS1...3). Read access to the header sections is done via OBF (RDHS1...3 + MBS). The data pointer has to be calculated by the programmer to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 33-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received **PLR[6:0]**, Receive Cycle Count **RCC[5:0]**, Received on Channel Indicator **RCI**, Startup Frame Indicator **SFI**, Sync Frame Indicator **SYN**, Null Frame Indicator **NFI**, Payload Preamble Indicator **PPI**, and Reserved Bit **RES** are updated from received valid data frames only.

Header word 3 of each configured message buffer holds the respective Message Buffer Status MBS.

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	P			M B I	T X M	P I T	C F G	C H B	C H A																								
1	P																																
2	P																																
3	P																																
...	P																																
...	P																																

	Frame Configuration
	Filter Configuration
	Message Buffer Control
	Message RAM Configuration
	Updated from received Data Frame
	Message Buffer Status MBS
	Parity Bit
	unused

Table 58.4-12 Header section of a message buffer in the Message RAM

MB91460 Series**Header 1** (word 0)

Write access via WRHS1, read access via RDHS1:

- Frame ID - Slot counter filtering configuration
- Cycle Code - Cycle counter filtering configuration
- CHA, CHB - Channel filtering configuration
- CFG - Message buffer direction configuration: receive / transmit
- PPIT - Payload Preamble Indicator Transmit
- TXM - Transmit mode configuration: single-shot / continuous
- MBI - Message buffer receive / transmit interrupt enable

Header 2 (word 1)

Write access via WRHS2, read access via RDHS2:

- Header CRC Transmit Buffer: Configured by the Host (calculated from frame header)
- Receive Buffer: Updated from received frame
- Payload Length Configured - Length of data section (2-byte words) as configured by the Host
- Payload Length Received - Length of payload segment (2-byte words) stored from received frame

Header 3 (word 2)

Write access via WRHS3, read access via RDHS3:

- Data Pointer- Pointer to the beginning of the corresponding data section in the data partition

Read access via RDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count - Cycle count from received frame
- RCI - Received on Channel Indicator
- SFI - Startup Frame Indicator
- SYN - Sync Frame Indicator
- NFI - Null Frame Indicator
- PPI - Payload Preamble Indicator
- RES - Reserved bit

Message Buffer Status MBS (word 3)

Read access via MBS, updated by the CC at the end of the configured slot.

- VFRA - Valid Frame Received on channel A
- VFRB - Valid Frame Received on channel B
- SEOA - Syntax Error Observed on channel A
- SEOB - Syntax Error Observed on channel B
- CEOA - Content Error Observed on channel A
- CEOB - Content Error Observed on channel B
- SVOA - Slot boundary Violation Observed on channel A
- SVOB - Slot boundary Violation Observed on channel B
- TCIA - Transmission Conflict Indication channel A
- TCIB - Transmission Conflict Indication channel B
- ESA - Empty Slot Channel A
- ESB - Empty Slot Channel B
- MLST - Message LoST
- FTA - Frame Transmitted on Channel A
- FTB - Frame Transmitted on Channel B
- Cycle Count Status - Actual cycle count when status was updated
- RCIS - Received on Channel Indicator Status
- SFIS - Startup Frame Indicator Status
- SYNS - Sync Frame Indicator Status
- NFIS - Null Frame Indicator Status
- PPIS - Payload Preamble Indicator Status
- RESS - Reserved bit Status

MB91460 Series

■ Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes plus one parity bit.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the programmer has to assure that the data pointers point to addresses within the data partition. [Table 58.4-13 Example for structure of the data partition in the Message RAM](#) below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see [Table 58.4-13](#) below).

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	P	unused								unused								unused								unused							
...	P	unused								unused								unused								unused							
...	P	MBn Data3								MBn Data2								MBn Data1								MBn Data0							
...	P							
...	P							
...	P	MBn Data(m)								MBn Data(m-1)								MBn Data(m-2)								MBn Data(m-3)							
...	P							
...	P							
...	P							
...	P	MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
...	P							
...	P	MB1 Data(k)								MB1 Data(k-1)								MB1 Data(k-2)								MB1 Data(k-3)							
2046	P	MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
2047	P	unused								unused								MB0 Data5								MB0 Data4							

Table 58.4-13 Example for structure of the data partition in the Message RAM

■ Parity Check

There is a parity checking mechanism implemented in the E-Ray core to assure the integrity of the data stored in the seven RAM blocks. The RAM blocks have a parity generator / checker attached as shown in figure 58.4-15. When data is written to a RAM block, the local parity generator generates the parity bit. The E-Ray core uses an even parity (with an even number of ones in the 32-bit data word a zero parity bit is generated). The parity bit is stored together with the respective data word. The parity is checked each time a data word is read from any of the RAM blocks. The E-Ray core's internal data buses have a width of 32 bits.

If a parity error is detected, the respective error flag is set. The parity error flags **MHDS.PIBF**, **MHDS.POBF**, **MHDS.PMR**, **MHDS.PTBF1**, **MHDS.PTBF2**, and the faulty message buffer indicators **MHDS.FMBD**, **MHDS.MFMB**, **MHDS.FMB[6:0]** are located in the Message Handler Status register. These single error flags control the error interrupt flag **EIR.PERR**.

Figure 58.4-15 shows the data paths between the RAM blocks and the parity generators / checkers.

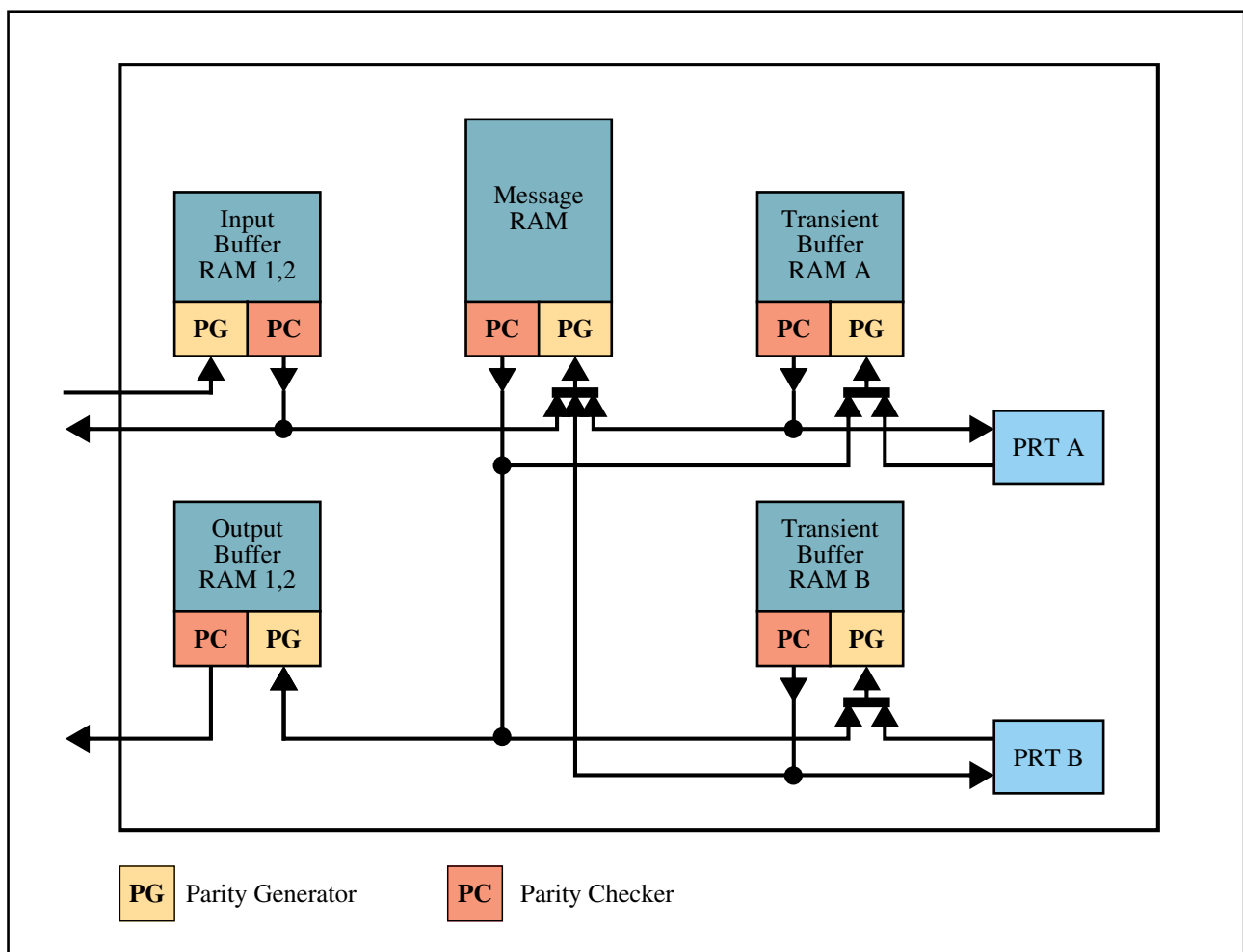


Figure 58.4-15 Parity generation and check

Note: Parity generator & checker are not part of the RAM blocks, but of the RAM access logic which is part of the E-Ray core.

MB91460 Series

When a parity error has been detected the following actions will be performed:

In all cases

- The respective parity error flag in register MHDS is set
- The parity error flag **EIR.PERR** is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases

1) Parity error during data transfer from Input Buffer RAM 1,2 \Rightarrow Message RAM

- a) Transfer of header and/or data section:
 - **MHDS.PIBF** bit is set
 - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
 - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
 - Transmit buffer: Transmission request for the respective message buffer is not set
- b) Transfer of data section only:

Parity error when reading header section of respective message buffer from Message RAM.

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- The data section of the respective message buffer is not updated
- Transmit buffer: Transmission request for the respective message buffer is not set

2) Parity error during Host reading Input Buffer RAM 1,2

- **MHDS.PIBF** bit is set

3) Parity error during scan of header sections in Message RAM

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- Ignore message buffer (message buffer is skipped)

4) Parity error during data transfer from Message RAM \Rightarrow Transient Buffer RAM 1, 2

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero.

5) Parity error during data transfer from Transient Buffer RAM 1, 2 \Rightarrow Protocol Controller 1, 2

- **MHDS.PTBF1,2** bit is set
- Frames already in transmission are invalidated by setting the frame CRC to zero

6) Parity error during data transfer from Transient Buffer RAM 1, 2 \Rightarrow Message RAM

- a) Parity error when reading header section of respective message buffer from Message RAM:

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- The data section of the respective message buffer is not updated

- b) Parity error when reading Transient Buffer RAM 1, 2:

- **MHDS.PTBF1,2** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer

• 7) Parity error during data transfer from Message RAM \Rightarrow Output Buffer RAM

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer

8) Parity error during Host reading Output Buffer RAM 1,2

- **MHDS.POBF** bit is set

• 9) Parity error during data read of Transient Buffer RAM 1, 2

When a parity error occurs when the Message Handler reads a frame with network management information (PPI = '1') from the Transient Buffer RAM 1, 2 the corresponding network management vector register NMV1...3 is not updated from that frame.

■ Host Handling of Parity Errors

Parity error caused by transient bit flips can be fixed by:

● Self-healing

Parity errors located in

- Input Buffer RAM 1,2
- Output Buffer RAM 1,2
- Data Section of Message RAM
- Transient Buffer RAM A
- Transient Buffer RAM B

are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

● CLEAR_RAM Command

When called in DEFAULT_CONFIG or CONFIG state POC command CLEAR_RAM initializes all module-internal RAMs to zero.

- Temporary Unlocking of Header Section

A parity error in the header section of a locked message buffer can be fixed by a transfer from the Input Buffer to the locked buffer Header Section. For this transfer, the write-access to the IBCR (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see 4.3.3 Lock Register (LCK)).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by MRC.SEC[1:0], and will be updated with new data.

58.4.13 Module Interrupt

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable / disable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers EIR and SIR.

Register	Bit	Function
EIR	PEMC	Protocol Error Mode Changed
	CNA	Command Not Valid
	SFBM	Sync Frames Below Minimum
	SFO	Sync Frame Overflow
	CCF	Clock Correction Failure
	CCL	CHI Command Locked
	PERR	Parity Error
	RFO	Receive FIFO Overrun
	EFA	Empty FIFO Access
	IIBA	Illegal Input Buffer Access
	IOBA	Illegal Output Buffer Access
	MHF	Message Handler Constraints Flag
	EDA	Error Detected on Channel A
	LTVA	Latest Transmit Violation Channel A
	TABA	Transmission Across Boundary Channel A
	EDB	Error Detected on Channel B
	LTVB	Latest Transmit Violation Channel B
	TABB	Transmission Across Boundary Channel B

MB91460 Series

Register	Bit	Function
SIR	WST	Wakeup Status
	CAS	Collision Avoidance Symbol
	CYCS	Cycle Start Interrupt
	TXI	Transmit Interrupt
	RXI	Receive Interrupt
	RFNE	Receive FIFO not Empty
	RFCL	Receive FIFO Critical Level
	NMVC	Network Management Vector Changed
	TI0	Timer Interrupt 0
	TI1	Timer Interrupt 1
	TIBC	Transfer Input Buffer Completed
	TOBC	Transfer Output Buffer Completed
	SWE	Stop Watch Event
	SUCS	Startup Completed Successfully
	MBSI	Message Buffer Status Interrupt
	SDS	Start of Dynamic Segment
	WUPA	Wakeup Pattern Channel A
	MTSA	MTS Received on Channel A
	WUPB	Wakeup Pattern Channel B
	MTSB	MTS Received on Channel B
ILE	EINT0	Enable Interrupt Line 0
	EINT1	Enable Interrupt Line 1

Table 58.4-14 Module interrupt flags and interrupt line enable

The interrupt lines to the Host, **FlexRay Channel 0** and **FlexRay Channel 1**, are controlled by the enabled interrupts. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit **ILE.EINT0** and **ILE.EINT1**.

The two timer interrupts generated by interrupt timer 0 and 1 are available on the interrupts **FlexRay Timer interrupt 0** and **FlexRay Timer interrupt 1**. They can be configured via registers T0C and T1C.

A stop watch event may be triggered via input pin **eray_stpwt**.

The status of the data transfer between IBF / OBF and the Message RAM is signalled on pins **eray_ibusy** and **eray_obusy**. When a transfer has completed bit **SIR.TIBC** or **SIR.TOBC** is set.

58.5. Appendix

58.5.1 Register Bit Overview

LCK		Lock Register															
0xD01C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1264	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EIR		Error Interrupt Register															
0xD020		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1265	R	0	0	0	0	0	TABB	LTVB	EDB	0	0	0	0	0	TABA	LTVA	EDA
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
SIR		Status Interrupt Register															
0xD024		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1268	R	0	0	0	0	0	0	MTSB	WUPB	0	0	0	0	0	0	MTSA	WUPA
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
EILS		Error Interrupt Line Select															
0xD028		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1271	R	0	0	0	0	0	TABBL	LTVBL	EDBL	0	0	0	0	0	TABAL	LTVAL	EDAL
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL

MB91460 Series

SILS		Status Interrupt Line Select															
0xD02C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1272	R	0	0	0	0	0	0	MTSBL	WUPBL	0	0	0	0	0	0	MTSAL	WUPAL
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
EIES EIER		Error Interrupt Enable Set Error Interrupt Enable Reset															
0xD030 0xD034		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1273	R	0	0	0	0	0	TABBE	LTVBE	EDBE	0	0	0	0	0	TABAE	LTVAE	EDAE
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
SIES SIER		Status Interrupt Enable Set Status Interrupt Enable Reset															
0xD038 0xD03C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1274	R	0	0	0	0	0	0	MTSBE	WUPBE	0	0	0	0	0	0	MTSAE	WUPAE
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
ILE		Interrupt Line Enable															
0xD040		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1275	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EINT1	EINT0
T0C		Timer 0 Configuration															
0xD044		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1276	R	0	0	T0MO13	T0MO12	T0MO11	T0MO10	T0MO9	T0MO8	T0MO7	T0MO6	T0MO5	T0MO4	T0MO3	T0MO2	T0MO1	T0MO0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	T0CC6	T0CC5	T0CC4	T0CC3	T0CC2	T0CC1	T0CC0	0	0	0	0	0	0	T0MS	T0RC
T1C		Timer 1 Configuration															
0xD048		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1277	R	0	0	T1MC13	T1MC12	T1MC11	T1MC10	T1MC9	T1MC8	T1MC7	T1MC6	T1MC5	T1MC4	T1MC3	T1MC2	T1MC1	T1MC0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T1MS	T1RC

STPW1		Stop Watch Register 1																
0xD04C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1278	R	0	0	SMTV13	SMTV12	SMTV11	SMTV10	SMTV9	SMTV8	SMTV7	SMTV6	SMTV5	SMTV4	SMTV3	SMTV2	SMTV1	SMTV0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	SCCV5	SCCV4	SCCV3	SCCV2	SCCV1	SCCV0	0	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT	
	W																	
STPW2		Stop Watch Register 2																
0xD050		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1279	R	0	0	0	0	0	SSCVB10	SSCVB9	SSCVB8	SSCVB7	SSCVB6	SSCVB5	SSCVB4	SSCVB3	SSCVB2	SSCVB1	SSCVB0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	SSCVA10	SSCVA9	SSCVA8	SSCVA7	SSCVA6	SSCVA5	SSCVA4	SSCVA3	SSCVA2	SSCVA1	SSCVA0	
	W																	
SUCC1		SUC Configuration Register 1																
0xD080		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1280	R	0	0	0	0	CCHB*		CCHA*	MTSB*	MTSA*	HCSE*	TSM*	WUCS*	PTA4*	PTA3*	PTA2*	PTA1*	PTA0*
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	CSA4*	CSA3*	CSA2*	CSA1*	CSA0*	0	TXSY*	TXST*	PBSY	0	0	0	CMD3		CMD2	CMD1	CMD0
	W																	
SUCC2		SUC Configuration Register 2																
0xD084		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1286	R	0	0	0	0	LTN3*		LTN2*	LTN1*	LTN0*	0	0	0	LT20*	LT19*	LT18*	LT17*	LT16*
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	LT15*	LT14*	LT13*	LT12*	LT11*	LT10*	LT9*	LT8*	LT7*	LT6*	LT5*	LT4*	LT3*	LT2*	LT1*	LT0*	
	W																	
SUCC3		SUC Configuration Register 3																
0xD088		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1286	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	0	0	0	WCF3*		WCF2*	WCF1*	WCF0*	WCP3*	WCP2*	WCP1*	WCP0*
	W																	
NEMC		NEM Configuration Register																
0xD08C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1287	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	0	0	0	0	0	0	0	0	NML3*		NML2*	NML1*	NML0*
	W																	

MB91460 Series

PRTC1		PRT Configuration Register 1																
0xD090		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1287	RW	RWP5*	RWP4*	RWP3*	RWP2*	RWP1*	RWP0*	0	RXW8*	RXW7*	RXW6*	RXW5*	RXW4*	RXW3*	RXW2*	RXW1*	RXW0*	
		15	14	13	12	11	10	9										8
	RW	BRP1*	BRP0*	SPP1*	SPP0*	0	CASM6	CASM5*	CASM4*	CASM3*	CASM2*	CASM1*	CASM0*	TSST3*	TSST2*	TSST1*	TSST0*	
PRTC2		PRT Configuration Register 2																
0xD094		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1288	RW	0	0	TXL5*	TXL4*	TXL3*	TXL2*	TXL1*	TXL0*	TXI7*	TXI6*	TXI5*	TXI4*	TXI3*	TXI2*	TXI1*	TXI0*	
		15	14															13
	RW	0	0	RXL5*	RXL4*	RXL3*	RXL2*	RXL1*	RXL0*	0	0	RXI5*	RXI4*	RXI3*	RXI2*	RXI1*	RXI0*	
MHDC		MHD Configuration Register																
0xD098		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1288	RW	0	0	0	SLT12*	SLT11*	SLT10*	SLT9*	SLT8*	SLT7*	SLT6*	SLT5*	SLT4*	SLT3*	SLT2*	SLT1*	SLT0*	
		15	14	13														12
	RW	0	0	0	0	0	0	0	0	0	0	SFDL6*	SFDL5*	SFDL4*	SFDL3*	SFDL2*	SFDL1*	SFDL0*
GTUC1		GTU Configuration Register 1																
0xD0A0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1290	RW	0	0	0	0	0	0	0	0	0	0	0	0	UT19*	UT18*	UT17*	UT16*	
		15	14	13	12	11	10	9	8	7	6	5	4					3
	RW	UT15*	UT14*	UT13*	UT12*	UT11*	UT10*	UT9*	UT8*	UT7*	UT6*	UT5*	UT4*	UT3*	UT2*	UT1*	UT0*	
GTUC2		GTU Configuration Register 2																
0xD0A4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1290	RW	0	0	0	0	0	0	0	0	0	0	0	0	SNM3*	SNM2*	SNM1*	SNM0*	
		15	14	13	12	11	10	9	8	7	6	5	4					3
	RW	0	0	MPC13*	MPC12*	MPC11*	MPC10*	MPC9*	MPC8*	MPC7*	MPC6*	MPC5*	MPC4*	MPC3*	MPC2*	MPC1*	MPC0*	
GTUC3		GTU Configuration Register 3																
0xD0A8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1291	RW	0	MIOB6*	MIOB5*	MIOB4*	MIOB3*	MIOB2*	MIOB1*	MIOB0*	0	MIOA6*	MIOA5*	MIOA4*	MIOA3*	MIOA2*	MIOA1*	MIOA0*	
		15								14								13
	RW	UIOB7*	UIOB6*	UIOB5*	UIOB4*	UIOB3*	UIOB2*	UIOB1*	UIOB0*	UIOA7*	UIOA6*	UIOA5*	UIOA4*	UIOA3*	UIOA2*	UIOA1*	UIOA0*	
GTUC4		GTU Configuration Register 4																
0xD0AC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1292	RW	0	0	OCS13*	OCS12*	OCS11*	OCS10*	OCS9*	OCS8*	OCS7*	OCS6*	OCS5*	OCS4*	OCS3*	OCS2*	OCS1*	OCS0*	
		15	14															13
	RW	0	0	NIT13*	NIT12*	NIT11*	NIT10*	NIT9*	NIT8*	NIT7*	NIT6*	NIT5*	NIT4*	NIT3*	NIT2*	NIT1*	NIT0*	

GTUC5		GTU Configuration Register 5															
0xD0B0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1292	R	DEC7*	DEC6*	DEC5*	DEC4*	DEC3*	DEC2*	DEC1*	DEC0*	0	0	0	CDD4*	CDD3*	CDD2*	CDD1*	CDD0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1292	R	DCB7*	DCB6*	DCB5*	DCB4*	DCB3*	DCB2*	DCB1*	DCB0*	DCA7*	DCA6*	DCA5*	DCA4*	DCA3*	DCA2*	DCA1*	DCA0*
	W																
GTUC6		GTU Configuration Register 6															
0xD0B4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1293	R	0	0	0	0	0	MOD10*	MOD9*	MOD8*	MOD7*	MOD6*	MOD5*	MOD4*	MOD3*	MOD2*	MOD1*	MOD0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1293	R	0	0	0	0	0	ASR10*	ASR9*	ASR8*	ASR7*	ASR6*	ASR5*	ASR4*	ASR3*	ASR2*	ASR1*	ASR0*
	W																
GTUC7		GTU Configuration Register 7															
0xD0B8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1294	R	0	0	0	0	0	0	NSS9*	NSS8*	NSS7*	NSS6*	NSS5*	NSS4*	NSS3*	NSS2*	NSS1*	NSS0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1294	R	0	0	0	0	0	0	SSL9*	SSL8*	SSL7*	SSL6*	SSL5*	SSL4*	SSL3*	SSL2*	SSL1*	SSL0*
	W																
GTUC8		GTU Configuration Register 8															
0xD0BC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1294	R	0	0	0	NMS12*	NMS11*	NMS10*	NMS9*	NMS8*	NMS7*	NMS6*	NMS5*	NMS4*	NMS3*	NMS2*	NMS1*	NMS0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1294	R	0	0	0	0	0	0	0	0	0	0	MSL5*	MSL4*	MSL3*	MSL2*	MSL1*	MSL0*
	W																
GTUC9		GTU Configuration Register 9															
0xD0C0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1295	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSI1*	DSI0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1295	R	0	0	0	MAPO4*	MAPO3*	MAPO2*	MAPO1*	MAPO0*	0	0	APO5*	APO4*	APO3*	APO2*	APO1*	APO0*
	W																
GTUC10		GTU Configuration Register 10															
0xD0C4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1295	R	0	0	0	0	0	MRC10*	MRC9*	MRC8*	MRC7*	MRC6*	MRC5*	MRC4*	MRC3*	MRC2*	MRC1*	MRC0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1295	R	0	0	MOC13*	MOC12*	MOC11*	MOC10*	MOC9*	MOC8*	MOC7*	MOC6*	MOC5*	MOC4*	MOC3*	MOC2*	MOC1*	MOC0*
	W																
GTUC11		GTU Configuration Register 11															
0xD0C8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1296	R	0	0	0	0	0	ERC2*	ERC1*	ERC0*	0	0	0	0	0	EOC2*	EOC1*	EOC0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P. 1296	R	0	0	0	0	0	0	ERCC1	ERCC0	0	0	0	0	0	0	EOCC1	EOCC0
	W																

MB91460 Series

CCSV		CC Status Vector															
0xD100		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1297	R	0	0	PSL5	PSL4	PSL3	PSL2	PSL1	PSL0	RCA4	RCA3	RCA2	RCA1	RCA0	WSV2	WSV1	WSV0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	CSI	CSAI	CSNI	0	0	SLM1	SLM0	HRQ	FSI	POCS5	POCS4	POCS3	POCS2	POCS1	POCS0
CCEV		CC Error Vector															
0xD104		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1300	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	PTAC4	PTAC3	PTAC2	PTAC1	PTAC0	ERRM1	ERRM0	0	0	CCFC3	CCFC2	CCFC1	CCFC0
SCV		Slot Counter Value															
0xD110		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1301	R	0	0	0	0	0	SCCB10	SCCB9	SCCB8	SCCB7	SCCB6	SCCB5	SCCB4	SCCB3	SCCB2	SCCB1	SCCB0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	SCCA10	SCCA9	SCCA8	SCCA7	SCCA6	SCCA5	SCCA4	SCCA3	SCCA2	SCCA1	SCCA0
MTCCV		Macrotick and Cycle Counter Value															
0xD114		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1301	R	0	0	0	0	0	0	0	0	0	0	CCV5	CCV4	CCV3	CCV2	CCV1	CCV0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	MTV13	MTV12	MTV11	MTV10	MTV9	MTV8	MTV7	MTV6	MTV5	MTV4	MTV3	MTV2	MTV1	MTV0
RCV		Rate Correction Value															
0xD118		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1302	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
OCV		Offset Correction Value															
0xD11C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1302	R	0	0	0	0	0	0	0	0	0	0	0	0	0	OCV18	OCV17	OCV16
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	OCV15	OCV14	OCV13	OCV12	OCV11	OCV10	OCV9	OCV8	OCV7	OCV6	OCV5	OCV4	OCV3	OCV2	OCV1	OCV0
SFS		Sync Frame Status															
0xD120		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1303	R	0	0	0	0	0	0	0	0	0	0	0	0	RCLR	MRCS	OCLR	MOCS
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	VSBO3	VSBO2	VSBO1	VSBO0	VSBE3	VSBE2	VSBE1	VSBE0	VSAO3	VSAO2	VSAO1	VSAO0	VSAE3	VSAE2	VSAE1	VSAE0

SWNIT		Symbol Window and NIT Status																
0xD124		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1304	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0	0	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA	
W																		
ACS		Aggregated Channel Status																
0xD128		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1306	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	0	0	0		SBVB	CIB	CEDB	SEDB	VFRB	0	0	0	SBVA	CIA	CEDA	SEDA	VFRA
W																		
ESIDn		Even Sync ID [1...15]																
0xD130 to 0xD168		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1308	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	RXEB	RXEA	0	0	0	0	EID9	EID8	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	
W																		
OSIDn		Odd Sync ID [1...15]																
0xD170 to 0xD1A8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1309	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	RXOB	RXOA	0	0	0	0	OID9	OID8	OID7	OID6	OID5	OID4	OID3	OID2	OID1	OID0	
W																		
NMVn		Network Management Vector [1...3]																
0xD1B0 to 0xD1B8		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1310	R	NM31	NM30	NM29	NM28	NM27	NM26	NM25	NM24	NM23	NM22	NM21	NM20	NM19	NM18	NM17	NM16	
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	NM15	NM14	NM13	NM12	NM11	NM10	NM9	NM8	NM7	NM6	NM5	NM4	NM3	NM2	NM1	NM0	
W																		
MRC		Message RAM Configuration																
0xD300		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
P. 1311	R	0	0	0	0	0		SPLM*	SEC1*	SEC0*	LCB7*	LCB6*	LCB5*	LCB4*	LCB3*	LCB2*	LCB1*	LCB0*
	W																	
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	R	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*	FDB0*	
W																		

MB91460 Series

FRF		FIFO Rejection Filter															
0xD304		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1313	R	0	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	FID10*	FID9*	FID8*	FID7*	FID6*	FID5*	FID4*	FID3*	FID2*	FID1*	FID0*	CH1*	CH0*
FRFM		FIFO Rejection Filter Mask															
0xD308		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1314	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	MFID10*	MFID9*	MFID8*	MFID7*	MFID6*	MFID5*	MFID4*	MFID3*	MFID2*	MFID1*	MFID0*	0	0
FCL		FIFO Critical Level															
0xD30C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1314	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	0	0	0	CL7*	CL6*	CL5*	CL4*	CL3*	CL2*	CL1*	CL0*
MHDS		Message Handler Status															
0xD310		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1315	R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM	MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
LDTS		Last Dynamic Transmit Slot															
0xD314		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1316	R	0	0	0	0	0	LDTB10	LDTB9	LDTB8	LDTB7	LDTB6	LDTB5	LDTB4	LDTB3	LDTB2	LDTB1	LDTB0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	LDTA10	LDTA9	LDTA8	LDTA7	LDTA6	LDTA5	LDTA4	LDTA3	LDTA2	LDTA1	LDTA0
FSR		FIFO Status Register															
0xD318		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1317	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	RFFL7	RFFL6	RFFL5	RFFL4	RFFL3	RFFL2	RFFL1	RFFL0	0	0	0	0	0	RFO	RFCL	RFNE
MHDF		Message Handler Constraints Flags															
0xD31C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1318	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	W	0	0	0	0	0	0	0	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA

TXRQ1		Transmission Request 1															
0xD320		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1320	R	TXR31	TXR30	TXR29	TXR28	TXR27	TXR26	TXR25	TXR24	TXR23	TXR22	TXR21	TXR20	TXR19	TXR18	TXR17	TXR16
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR15	TXR14	TXR13	TXR12	TXR11	TXR10	TXR9	TXR8	TXR7	TXR6	TXR5	TXR4	TXR3	TXR2	TXR1	TXR0
	W																
TXRQ2		Transmission Request 2															
0xD324		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1320	R	TXR63	TXR62	TXR61	TXR60	TXR59	TXR58	TXR57	TXR56	TXR55	TXR54	TXR53	TXR52	TXR51	TXR50	TXR49	TXR48
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR47	TXR46	TXR45	TXR44	TXR43	TXR42	TXR41	TXR40	TXR39	TXR38	TXR37	TXR36	TXR35	TXR34	TXR33	TXR32
	W																
TXRQ3		Transmission Request 3															
0xD328		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1320	R	TXR95	TXR94	TXR93	TXR92	TXR91	TXR90	TXR89	TXR88	TXR87	TXR86	TXR85	TXR84	TXR83	TXR82	TXR81	TXR80
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR79	TXR78	TXR77	TXR76	TXR75	TXR74	TXR73	TXR72	TXR71	TXR70	TXR69	TXR68	TXR67	TXR66	TXR65	TXR64
	W																
TXRQ4		Transmission Request 4															
0xD32C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1320	R	TXR127	TXR126	TXR125	TXR124	TXR123	TXR122	TXR121	TXR120	TXR119	TXR118	TXR117	TXR116	TXR115	TXR114	TXR113	TXR112
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	TXR111	TXR110	TXR109	TXR108	TXR107	TXR106	TXR105	TXR104	TXR103	TXR102	TXR101	TXR100	TXR99	TXR98	TXR97	TXR96
	W																
NDAT1		New Data 1															
0xD330		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1321	R	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
	W																
NDAT2		New Data 2															
0xD334		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1321	R	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	W																
NDAT3		New Data 3															
0xD338		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1321	R	ND95	ND94	ND93	ND92	ND91	ND90	ND89	ND88	ND87	ND86	ND85	ND84	ND83	ND82	ND81	ND80
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND79	ND78	ND77	ND76	ND75	ND74	ND73	ND72	ND71	ND70	ND69	ND68	ND67	ND66	ND65	ND64
	W																

MB91460 Series

NDAT4		New Data 4															
0xD33C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1321	R	ND127	ND126	ND125	ND124	ND123	ND122	ND121	ND120	ND119	ND118	ND117	ND116	ND115	ND114	ND113	ND112
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ND111	ND110	ND109	ND108	ND107	ND106	ND105	ND104	ND103	ND102	ND101	ND100	ND99	ND98	ND97	ND96
W																	
MBSC1		Message Buffer Status Changed 1															
0xD340		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1322	R	MBC31	MBC30	MBC29	MBC28	MBC27	MBC26	MBC25	MBC24	MBC23	MBC22	MBC21	MBC20	MBC19	MBC18	MBC17	MBC16
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC15	MBC14	MBC13	MBC12	MBC11	MBC10	MBC9	MBC8	MBC7	MBC6	MBC5	MBC4	MBC3	MBC2	MBC1	MBC0
W																	
MBSC2		Message Buffer Status Changed 2															
0xD344		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1322	R	MBC63	MBC62	MBC61	MBC60	MBC59	MBC58	MBC57	MBC56	MBC55	MBC54	MBC53	MBC52	MBC51	MBC50	MBC49	MBC48
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC47	MBC46	MBC45	MBC44	MBC43	MBC42	MBC41	MBC40	MBC39	MBC38	MBC37	MBC36	MBC35	MBC34	MBC33	MBC32
W																	
MBSC3		Message Buffer Status Changed 3															
0xD348		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1322	R	MBC95	MBC94	MBC93	MBC92	MBC91	MBC90	MBC89	MBC88	MBC87	MBC86	MBC85	MBC84	MBC83	MBC82	MBC81	MBC80
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC79	MBC78	MBC77	MBC76	MBC75	MBC74	MBC73	MBC72	MBC71	MBC70	MBC69	MBC68	MBC67	MBC66	MBC65	MBC64
W																	
MBSC4		Message Buffer Status Changed 4															
0xD34C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1322	R	MBC127	MBC126	MBC125	MBC124	MBC123	MBC122	MBC121	MBC120	MBC119	MBC118	MBC117	MBC116	MBC115	MBC114	MBC113	MBC112
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MBC111	MBC110	MBC109	MBC108	MBC107	MBC106	MBC105	MBC104	MBC103	MBC102	MBC101	MBC100	MBC99	MBC98	MBC97	MBC96
W																	
CREL		Core Release Register															
0xD3F0		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1323	R	REL3	REL2	REL1	REL0	STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	STEP0	YEAR3	YEAR2	YEAR1	YEAR0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MON7	MON6	MON5	MON4	MON3	MON2	MON1	MON0	DAY7	DAY6	DAY5	DAY4	DAY3	DAY2	DAY1	DAY0
W																	
ENDN		Endian Register															
0xD3F4		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1324	R	ETV31	ETV30	ETV29	ETV28	ETV27	ETV26	ETV25	ETV24	ETV23	ETV22	ETV21	ETV20	ETV19	ETV18	ETV17	ETV16
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	ETV15	ETV14	ETV13	ETV12	ETV11	ETV10	ETV9	ETV8	ETV7	ETV6	ETV5	ETV4	ETV3	ETV2	ETV1	ETV0
W																	

WRDSn		Write Data Section [1...64]															
0xD400 to 0xD4FC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1324	RW	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RW	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
WRHS1		Write Header Section 1															
0xD500		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1325	RW	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	RW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
WRHS2		Write Header Section 2															
0xD504		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1326	RW	0	0	0	0	0	0	0	0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
	RW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
WRHS3		Write Header Section 3															
0xD508		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1327	RW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0	0	0	DP10*	DP9*	DP8*	DP7*	DP6*	DP5*	DP4*	DP3*	DP2*	DP1*	DP0*
IBCM		Input Buffer Command Mask															
0xD510		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1327	RW	0	0	0	0	0	0	0	0	0	0	0	0	0	STXRS	LDSS	LHSS
	RW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	STXRH	LDSH	LHSH
IBCR		Input Buffer Command Request															
0xD514		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1328	RW	IBSYS	0	0	0	0	0	0	0	0	IBRS6	IBRS5	IBRS4	IBRS3	IBRS2	IBRS1	IBRS0
	RW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		IBSYH	0	0	0	0	0	0	0	0	IBRH6	IBRH5	IBRH4	IBRH3	IBRH2	IBRH1	IBRH0

MB91460 Series

RDDSn		Read Data Section [1...64]															
0xD600 to 0xD6FC		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1329	R	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	W																
RDHS1		Read Header Section 1															
0xD700		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1330	R	0	0	MBI	TXM	PPIT	CFG	CHB	CHA	0	CYC6	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	FID10	FID9	FID8	FID7	FID6	FID5	FID4	FID3	FID2	FID1	FID0
	W																
RDHS2		Read Header Section 2															
0xD704		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1331	R	0	PLR6	PLR5	PLR4	PLR3	PLR2	PLR1	PLR0	0	PLC6	PLC5	PLC4	PLC3	PLC2	PLC1	PLC0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	CRC10	CRC9	CRC8	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0
	W																
RDHS3		Read Header Section 3															
0xD708		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1332	R	0	0	RES	PPI	NFI	SYN	SFI	RCI	0	0	RCC5	RCC4	RCC3	RCC2	RCC1	RCC0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	DP10	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
	W																
MBS		Message Buffer Status															
0xD70C		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1333	R	0	0	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	0	0	CCS5	CCS4	CCS3	CCS2	CCS1	CCS0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	FTB	FTA	0	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	W																
OBCM		Output Buffer Command Mask															
0xD710		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1336	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RDSH	RHSH
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R															RDSS	RHSS
	W																
OBCR		Output Buffer Command Request															
0xD714		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P. 1337	R	0	0	0	0	0	0	0	0	0	OBRH6	OBRH5	OBRH4	OBRH3	OBRH2	OBRH1	OBRH0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	OBSYS	0	0	0	0	0		REQ	VIEW		OBR6	OBR5	OBR4	OBR3	OBR2	OBR1
	W																

Table 58.5-1 Register bit overview

58.5.2 Assignment of FlexRay Configuration Parameters

Parameter	Bit(field)	Page
pKeySlotUsedForStartup	SUCC1.TXST	P. 1280
pKeySlotUsedForSync	SUCC1.TXSY	P. 1280
gColdStartAttempts	SUCC1.CSA[4:0]	P. 1280
pAllowPassiveToActive	SUCC1.PTA[4:0]	P. 1280
pWakeupChannel	SUCC1.WUCS	P. 1280
pSingleSlotEnabled	SUCC1.TSM	P. 1280
pAllowHaltDueToClock	SUCC1.HCSE	P. 1280
pChannels	SUCC1.CCHA SUCC1.CCHB	P. 1280
pdListenTimeOut	SUCC2.LT[20:0]	P. 1286
gListenNoise	SUCC2.LTN[3:0]	P. 1286
gMaxWithoutClockCorrectionPassive	SUCC3.WCP[3:0]	P. 1286
gMaxWithoutClockCorrectionFatal	SUCC3.WCF[3:0]	P. 1286
gNetworkManagementVectorLength	NEMC.NML[3:0]	P. 1287
gdTSSTransmitter	PRTC1.TSST[3:0]	P. 1287
gdCASRxLowMax	PRTC1.CASM[6:0]	P. 1287
gdSampleClockPeriod	PRTC1.BRP[1:0]	P. 1287
pSamplesPerMicrotick	PRTC1.BRP[1:0]	P. 1287
gdWakeupSymbolRxWindow	PRTC1.RXW[8:0]	P. 1287
pWakeupPattern	PRTC1.RWP[5:0]	P. 1287
gdWakeupSymbolRxIdle	PRTC2.RXI[5:0]	P. 1288
gdWakeupSymbolRxLow	PRTC2.RXL[5:0]	P. 1288
gdWakeupSymbolTxIdle	PRTC2.TXI[7:0]	P. 1288
gdWakeupSymbolTxLow	PRTC2.TXL[5:0]	P. 1288
gPayloadLengthStatic	MHDC.SFDL[6:0]	P. 1288
pLatestTx	MHDC.SLT[12:0]	P. 1288
pMicroPerCycle	GTUC1.UT[19:0]	P. 1290
gMacroPerCycle	GTUC2.MPC[13:0]	P. 1290
gSyncNodeMax	GTUC2.SNM[3:0]	P. 1290
pMicroInitialOffset[A]	GTUC3.UIOA[7:0]	P. 1291
pMicroInitialOffset[B]	GTUC3.UIOB[7:0]	P. 1291
pMacroInitialOffset[A]	GTUC3.MIOA[6:0]	P. 1291

MB91460 Series

Parameter	Bit(field)	Page
pMacroInitialOffset[B]	GTUC3.MIOB[6:0]	P. 1291
gdNIT	GTUC4.NIT[13:0]	P. 1292
gOffsetCorrectionStart	GTUC4.OCs[13:0]	P. 1292
pDelayCompensation[A]	GTUC5.DCA[7:0]	P. 1292
pDelayCompensation[B]	GTUC5.DCB[7:0]	P. 1292
pClusterDriftDamping	GTUC5.CDD[4:0]	P. 1292
pDecodingCorrection	GTUC5.DEC[7:0]	P. 1292
pdAcceptedStartupRange	GTUC6.ASR[10:0]	P. 1293
pdMaxDrift	GTUC6.MOD[10:0]	P. 1293
gdStaticSlot	GTUC7.SSL[9:0]	P. 1294
gNumberOfStaticSlots	GTUC7.NSS[9:0]	P. 1294
gdMinislot	GTUC8.MSL[5:0]	P. 1294
gNumberOfMinislots	GTUC8.NMS[12:0]	P. 1294
gdActionPointOffset	GTUC9.APO[5:0]	P. 1295
gdMinislotActionPointOffset	GTUC9.MAPO[4:0]	P. 1295
gdDynamicSlotIdlePhase	GTUC9.DSI[1:0]	P. 1295
pOffsetCorrectionOut	GTUC10.MOC[13:0]	P. 1295
pRateCorrectionOut	GTUC10.MRC[10:0]	P. 1295
pExternOffsetCorrection	GTUC11.EOC[2:0]	P. 1296
pExternRateCorrection	GTUC11.ERC[2:0]	P. 1296

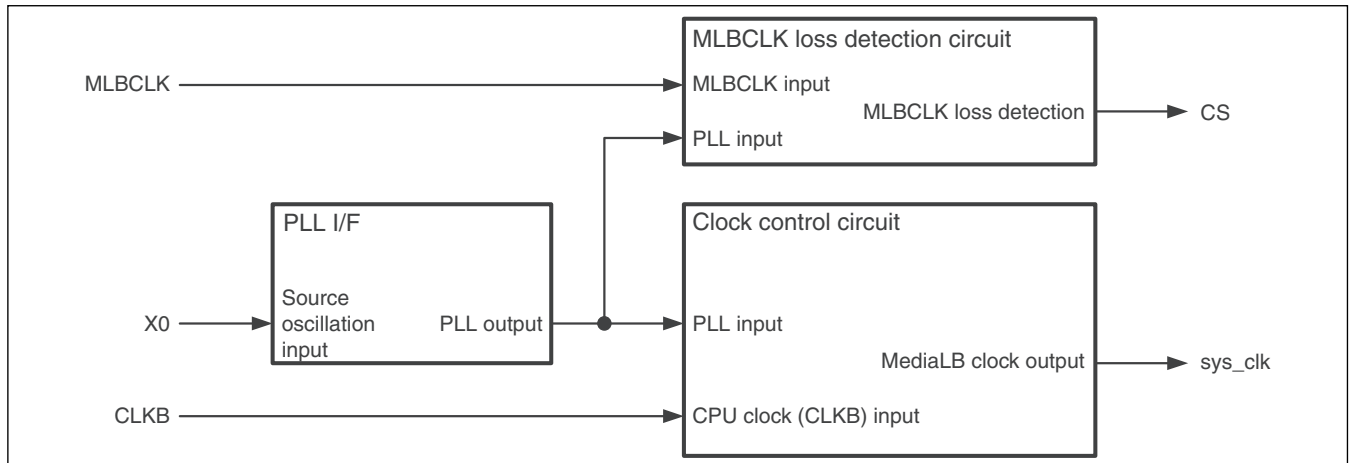
Table 58.5-2 FlexRay configuration parameters

Chapter 59 MediaLB Clock Generation/Bus Interface

59.1. MediaLB clock generation block

59.1.1 Overview

The MediaLB clock generation block consists of PLL-I/F, a clock control circuit, and a MLBCLK loss detection circuit.



CS : Clock stop detection signal

sys_clk : MediaLB macro operation clock

● PLL-I/F

This circuit generates a high-frequency clock by multiplying the source oscillation.

It is used for the clock control circuit and the MLBCLK loss detection circuit, described in later paragraphs.

It has a built-in PLL especially designed for MediaLB. PLL's output frequency range is from 100 to 224MHz.

It has a clock gear function to prevent rapid load changes.

● Clock control circuit

This circuit generates the MediaLB macro operation clock using a high-speed clock from PLL-I/F.

The circuit has a prescaler function to divide the high-speed clock from PLL-I/F.

Available division ratios range from no division to 16 divisions.

It also has a clock switch over function to select the MediaLB macro operation clock from the CPU clock or the clock from PLL-I/F.

At reset, the CPU clock (CLKB) divided by 2 is selected.

● MLBCLK loss detection circuit

This circuit uses the high-speed clock from PLL-I/F to monitor the MediaLB clock (MLBCLK) which is supplied from the MediaLB bus master.

An interrupt can be generated when MLBCLK goes below a certain frequency.

This allows abnormal stop of MLBCLK to be detected.

59.1.2 MediaLB macro operation clock

As the macro operation clock, a high-speed clock is required which is more than twice as fast as MLBCLK supplied from the bus master.

For exact frequencies of MLBCLK, see "Media Local Bus Specification 3.0".

Recommended macro operation frequencies are as follows.

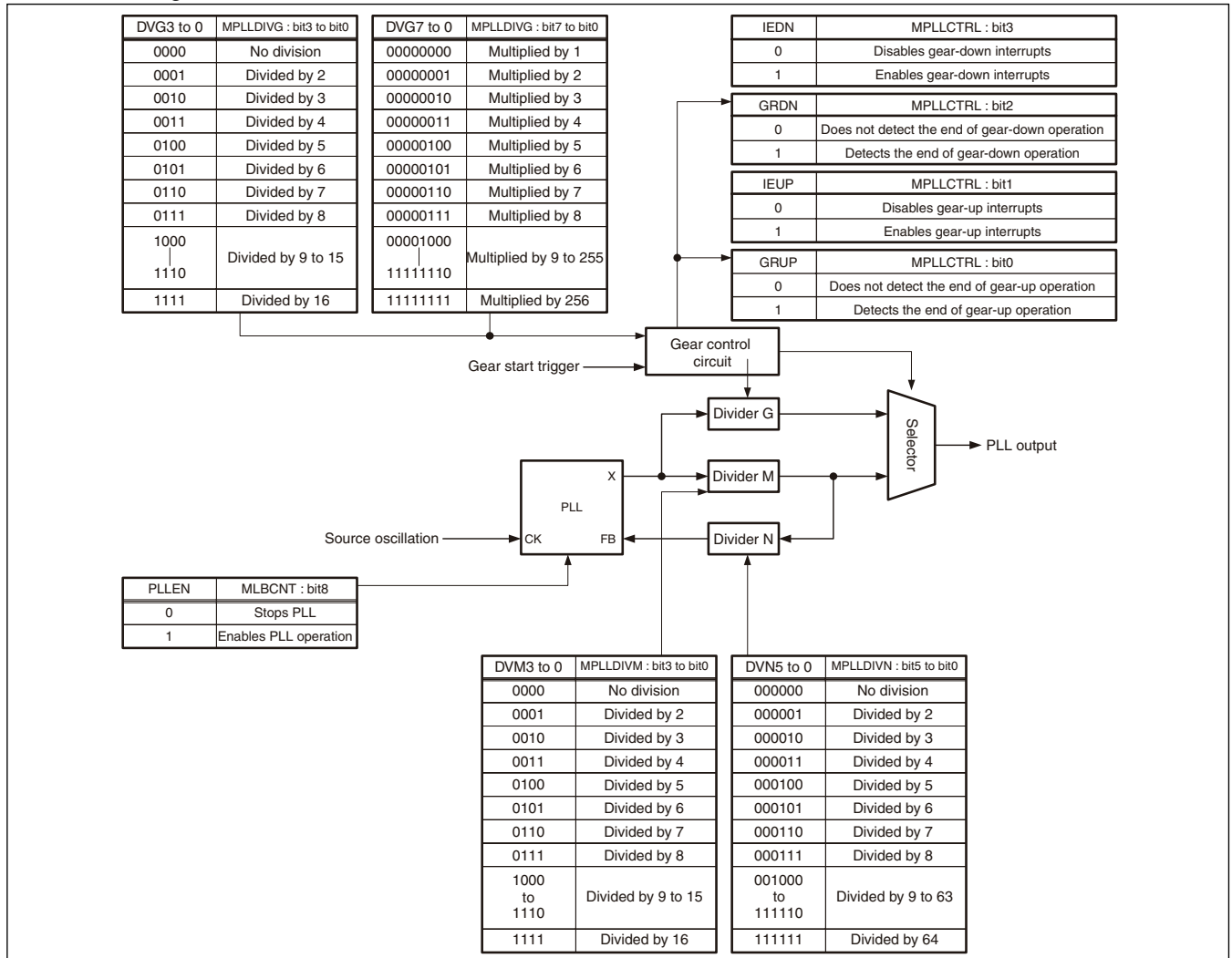
MLBCLK frequency	Recommended macro operation clock frequency
512Fs (Fs=48kHz)	64MHz or more
256Fs (Fs=48kHz)	32MHz or more

59.2. PLL interface

59.2.1 Overview

This block consists of PLL, division circuits, and a clock selector.

Its block diagram is shown below.



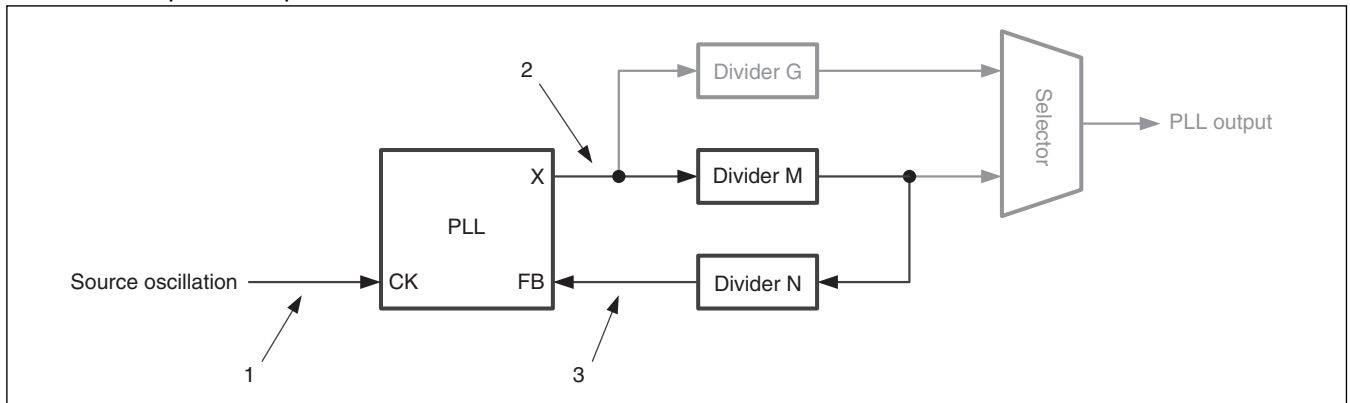
Note:

Source oscillation:	Input clock from X0
PLL output:	PLL interface output clock
PLL :	PLL
Gear control circuit:	Block which controls gear operation
Gear start trigger:	Trigger which indicates the start of gear operation
Divider G:	Divider for gear operation
Divider M:	Divider for PLL multiplication setting 1
Divider N:	Divider for PLL multiplication setting 2
Selector :	Divider G clock/Divider M clock selector

"PLL output" is the clock to be input into the MediaLB clock control circuit and the MLBCLK loss detection circuit.

59.2.2 Multiplication

The concept of multiplication is as follows.



The frequency of PLL-VCO output (Clock 2) is controlled so that the rising phase of Clock 1 matches that of Clock 3 shown in the above diagram.

For example, when Clock 1 is set to 4MHz and $M \times N$ to 40, Clock 3 is locked at 4MHz. Consequently, Clock 2 is set to " $4 \times 40 = 160\text{MHz}$ ". This means a 4MHz $M \times N$ clock.

A desired multiplication rate can be set by using Dividers M and N appropriately.

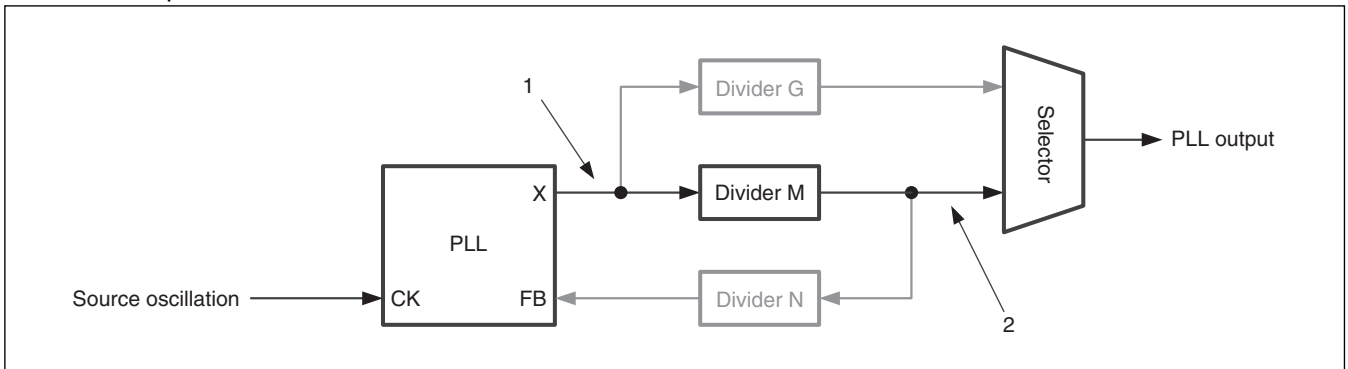
In the above example, there are several combinations to achieve " $M \times N = 40$ ". Therefore, there are more than one combination that can generate a multiply-by-40 clock.

M and N are set using the M PLLDIVM and M PLLDIVN registers, respectively.

M can be set within the range from 1 to 16, while N can be set within the range from 1 to 64.

59.2.3 PLL

The concept of PLL is as follows.



PLL-VCO output (Clock 1) is divided by M to become PLL output (Clock 2).

For example, when Clock 1 is set to 160MHz and M to 2, PLL output is 80MHz.

The selector will be explained later.

The degree of freedom for combinations of Divider M and Divider N, which determine PLL's multiplication rate, is high. However, select it by adding this final PLL output frequency.

For example, the PLL output frequency for "M=2, N=20" is different from that of "M=20, N=2", while both are multiplied to equal 40.

Although M can also be set to an odd number, the duty ratio of PLL output does not become 50:50.

The same applies to the case where no division is performed.

If the duty ratio is not 50:50, make sure to perform division using the clock control circuit described in later paragraphs.

Since the MLBCLK loss detection circuit always divides this PLL output when using it, division by an odd number causes no problem.

59.2.4 Gear

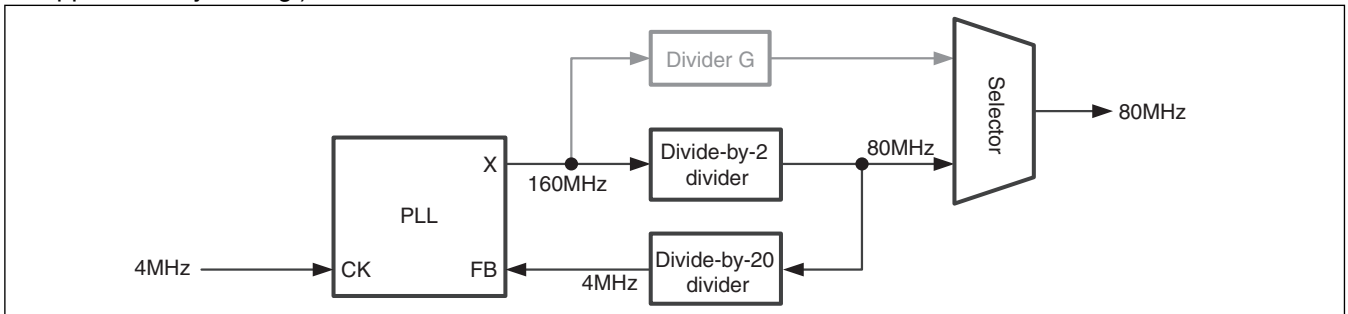
This function changes PLL output in stages as a measure to prevent a drop in the internal voltage due to a rapid load change.

The clock gear starts up, when the clock source is switched by the clock control circuit described in later paragraphs.

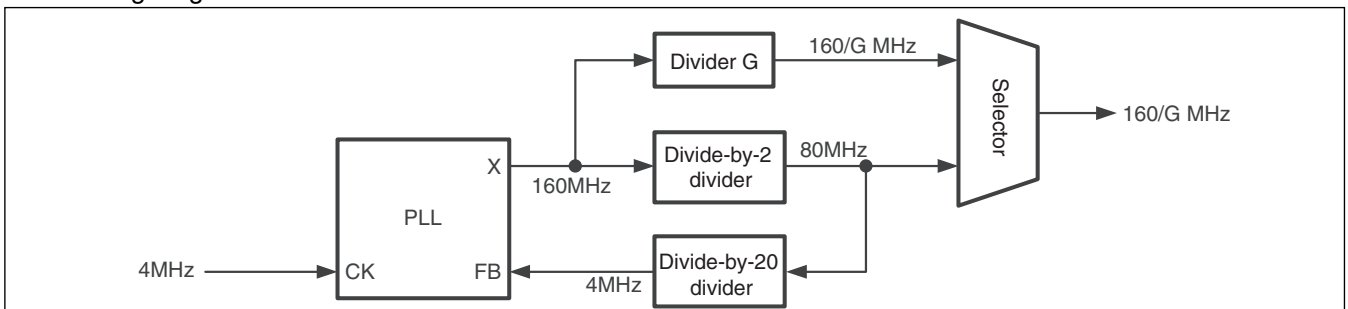
It must be set before the clock source is switched.

The following diagrams show an example assuming that the source oscillation is 4MHz, multiplied by 40, and the required PLL output is 80MHz.

Assume $M=2/N=20$. (Although certain values are provided in the diagrams for reference, the same concept applies to any setting.)



During the activation of the clock gear, the output of Divider G becomes the PLL output, as shown in the following diagram.



Division ratio G of Divider G is controlled by hardware.

The initial value can be set using the MPLLDIVG register.

The initial value can be set within the range from 1 to 16.

For example, when the initial value of G is set to 16:

Divider G starts from 16 divisions, which will be then subtracted by hardware to become M divisions in the end.

In this example, the PLL output starts from (160/16)MHz and then gradually speeds up to (160/2)MHz.

Once G equals M, the clock selected is switched automatically to the clock of Divider M.

In other words, $G > M$ must be achieved.

While this is an example of switching from a low-speed clock to a high-speed clock, the reverse also functions in the same way.

In that case, values from M to the initial value (MPLLDIVG) are added to G.

If the clock is switched by the clock control circuit, a gear operation start trigger is set to start gear operation.

When the gear operation is completed, the gear down/up end flag bit is set.

If interrupts are enabled, an interrupt occurs.

This interrupt is placed on #141 PLL Clock Gear interrupt.

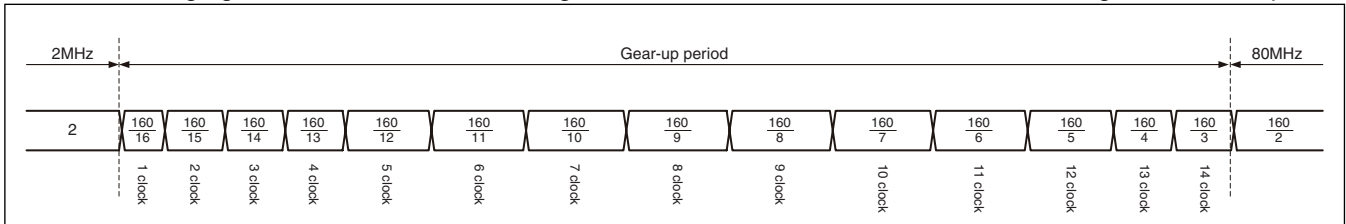
Note:

It must be noted that the MediaLB clock can be switched from CLKB. A sudden clock change may occur, depending on the CLKB frequency for when this function is used and the gear start frequency.

59.2.5 Time required for gear shifting

When the clock gear is used, the clock frequency of PLL output changes in stages. For this reason, a desired frequency cannot be achieved immediately after the PLL clock is selected by the clock control circuit described in later paragraphs.

The following figure shows a transition timing chart for when the source oscillation is changed to PLL output.



Source oscillation: 4MHz

G : 16

M : 2

Final PLL output: 80MHz

* M PLLMULG=1 is assumed. M PLLMULG will be explained later.

In this case, the PLL output shifts in stages, from 2MHz to (160/16)MHz, (160/15)MHz and finally to 160/2=80MHz.

The time spent on each frequency increases on a clock-by-clock basis in order to even it out as much as possible: one clock for (160/16)MHz, two clocks for (160/15)MHz, etc. As a result, the chart in the above figure can be established.

The total length can be extended by setting DLG[7:0] (M PLLMULG).

While it is as shown above for DLG[7:0]=8'h01, it doubles in total for DLG[7:0]=8'h02.

It can be extended up to 256 times.

Ineffective clock gear operation may be avoided by extending the time.

The actual total time can be expressed in the following formula.

$$\text{Gear-up/down time} = \text{M PLLMULG} \times T \times \left(\sum_{k=1}^G k \times (G - k + 1) - \sum_{k=G-M+1}^G k \times (G - k + 1) \right)$$

Total PLL-VCO output clock count for gear-up operation from G to M

Total PLL-VCO output clock count for gear-up operation from G to 1

Total PLL-VCO output clock count for gear-up operation from M - 1 to 1

T : PLL-VCO output cycle

59.2.6 Registers

■ MPLLDIVM:PLL Divider M

The following figure shows the configuration of PLL Divider M.

MPLLDIVM	bit	15	14	13	12	11	10	9	8
Address:0004F4 _H		-	-	-	-	DVM3	DVM2	DVM1	DVM0
	R/W	-	-	-	-	R/W	R/W	R/W	R/W
Initial value (INITX or WD-reset)		-	-	-	-	0	0	0	0
Initial value (Software reset)		-	-	-	-	X	X	X	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register sets a division ratio for Divider M.

It is one of the values used to determine PLL's multiplication rate as well as the value used to divide PLL VCO output and determine the final PLL output clock frequency.

[bit7,bit6]: Reserved bits

These bits are reserved.

"0" is always read.

Writing does not affect the operation.

[bit5 to bit0] DVM3-0: Divider M division ratio setting bits

Division ratios are as follows.

DVM3 to 0	Division ratio
0000 _B	No division
0001 _B	Divided by 2
0010 _B	Divided by 3
0011 _B	Divided by 4
0100 _B	Divided by 5
0101 _B	Divided by 6
0110 _B	Divided by 7
0111 _B	Divided by 8
1000 _B 1110 _B	Divided by 9 to 15
1111 _B	Divided by 16

Notes:

- Although M can also be set to an odd number, the duty ratio of PLL output does not become 50:50. The same applies to the case where no division is performed.
If the duty ratio is not 50:50, make sure to perform division using the clock control circuit described in later paragraphs.
- The value of this register must not be changed, when PLL is selected as the clock source.

■ MPLLDIVM: PLL Divider N

The following figure shows the configuration of PLL Divider N.

MPLLDIVM	bit	7	6	5	4	3	2	1	0
Address:0004F5 _H		-	-	DVN5	DVN4	DVN3	DVN2	DVN1	DVN0
R/W		-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INITX or WD-reset)		-	-	0	0	0	0	0	0
Initial value (Software reset)		-	-	X	X	X	X	X	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register sets a division ratio for Divider N.

It is one of the values used to determine PLL's multiplication rate.

[bit7,bit6]: Reserved bits

These bits are reserved.

The read value is "0".

Writing does not affect the operation.

[bit5 to bit0] DVN5 to 0: Division ratio N setting bits

Division ratios are as follows.

DVM5 to 0	Division ratio
000000 _B	No division
000001 _B	Divided by 2
000010 _B	Divided by 3
000011 _B	Divided by 4
000100 _B	Divided by 5
000101 _B	Divided by 6
000110 _B	Divided by 7
000111 _B	Divided by 8
001000 _B 111110 _B	Divided by 9 to 63
111111 _B	Divided by 64

Note:

- The value of this register must not be changed, when PLL is selected as the clock source.

■ MPLLDIVG: Clock gear initial/final division value setting register

The following figure shows the configuration of the clock gear initial division value setting register.

MPLLDIVG	bit	15	14	13	12	11	10	9	8
Address:0004F6 _H		-	-	-	-	DVG3	DVG2	DVG1	DVG0
R/W		-	-	-	-	R/W	R/W	R/W	R/W
Initial value (INITX or WD-reset)		-	-	-	-	0	0	0	0
Initial value (Software reset)		-	-	-	-	X	X	X	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register sets an initial/final division ratio for Divider G.

For clock gear-up operation, it becomes the initial division value.

For clock gear-down operation, it becomes the final division value.

[bit15 to bit12]: Reserved bits

These bits are reserved.

The read value is "0".

Writing does not affect the operation.

[bit11 to bit8] DVG3 to 0: Division ratio G setting bits

Division ratios are as follows.

DVG3 to 0	Division ratio
0000 _B	No division
0001 _B	Divided by 2
0010 _B	Divided by 3
0011 _B	Divided by 4
0100 _B	Divided by 5
0101 _B	Divided by 6
0110 _B	Divided by 7
0111 _B	Divided by 8
1000 _B 1110 _B	Divided by 9 to 15
1111 _B	Divided by 16

Note:

- The value of this register must not be changed, when PLL is selected as the clock source.

■ MPLLMULG: Clock gear cycle extension register

The following figure shows the configuration of the clock gear cycle extension register.

MPLLMULG	bit	7	6	5	4	3	2	1	0
Address:0004F7 _H		DLG7	DLG6	DLG5	DLG4	DLG3	DLG2	DLG1	DLG0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INITX or WD-reset)		0	0	0	0	0	0	0	0
Initial value (Software reset)		X	X	X	X	X	X	X	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register is used to extend the time from the start to the end of clock gear operation.

[bit11 to bit8] DLG7 to 0: Clock gear time magnification setting bits

Magnifications are as follows.

DLG7 to 0	Division ratio
00000000 _B	No division
00000001 _B	Divided by 2
00000010 _B	Divided by 3
00000011 _B	Divided by 4
00000100 _B	Divided by 5
00000101 _B	Divided by 6
00000110 _B	Divided by 7
00000111 _B	Divided by 8
00001000 _B 11111110 _B	Divided by 9 to 255
11111111 _B	Divided by 256

Note:

- The value of this register must not be changed, when PLL is selected as the clock source.

■ MPLLCTRL: Clock gear control register

The following figure shows the configuration of the clock gear control register.

MPLLCTRL	bit	15	14	13	12	11	10	9	8
Address:0004F8 _H		-	-	-	-	IEDN	GRDN	IEUP	GRUP
R/W		-	-	-	-	R/W	R/W	R/W	R/W
Initial value (INITX or WD-reset)		-	-	-	-	0	0	0	0
Initial value (Software reset)		-	-	-	-	X	X	X	X
WD-reset : Watchdog reset Initialization X: Not initialized									

This register has a flag to indicate the end of clock gear operation and controls the generation of interrupts.

[bit7 to bit4]: Reserved bits

These bits are reserved.

"0" is always read. Always write "0".

[bit3]: Gear-down interrupt enable bit

IEDN	Function
0	Disables gear-down interrupts [initial value]
1	Enables gear-down interrupts

An interrupt occurs, if MPLLCToRL:GRDN is set to "1" when "1" is written to this bit.

[bit2]: Gear-down end flag

GRDN	Function
0	Does not detect the end of gear-down operation [initial value]
1	Detects the end of gear-down operation

When this bit is set to "1" with MPLLCTRL:IEDN=1, an end-of-gear-down interrupt occurs.

Writing "1" to this bit has no meaning. Write "0" to clear the flag. "1" is read by RMW instruction.

[bit1]: Gear-up interrupt enable bit

IEUP	Function
0	Disables gear-up interrupts [initial value]
1	Enables gear-up interrupts

An interrupt occurs, if MPLLCTRL:GRUP is set to "1" when "1" is written to this bit.

[bit0]: Gear-up end flag

GRUP	Function
0	Does not detect the end of gear-up operation [initial value]
1	Detects the end of gear-up operation

When this bit is set to "1" with MPLLCTRL:IEUP=1, an end-of-gear-up interrupt occurs.

Writing "1" to this bit has no meaning. Write "0" to clear the flag. "1" is read by RMW instruction.

■ MLBCNT: MediaLB control register

The following figure shows the configuration of the MediaLB control register.

MLBCNT	bit	15	14	13	12	11	10	9	8
Address:0004FA _H		CCLKSEL	INTE	CS	-	-	-	-	PLLEN
	R/W	R/W	R/W	R	-	-	-	-	R/W
Initial value (INITX or WD-reset)		0	0	0	-	-	-	-	0
Initial value (Software reset)		X	X	X	-	-	-	-	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register is used to enable PLL operation.

It is also used as a setting register for clock loss detection.

[bit15] CCLKSEL: MLBCLK monitoring clock division setting bit

For details, see the chapter about the MLBCLK loss detection circuit.

[bit14] INTE: MLBCLK loss detection interrupt enable bit

For details, see the chapter about the MLBCLK loss detection circuit.

[bit13] CS: MLBCLK loss detection bit

For details, see the chapter about the MLBCLK loss detection circuit.

[bit12 to bit9]: Reserved bits

These bits are reserved.

"0" is always read.

Writing does not affect the operation.

[bit8] PLLEN: MediaLB PLL operation enable bit

This bit enables PLL operation for MediaLB.

PLLEN	PLL operation
0	Does not enable the operation
1	Enables the operation

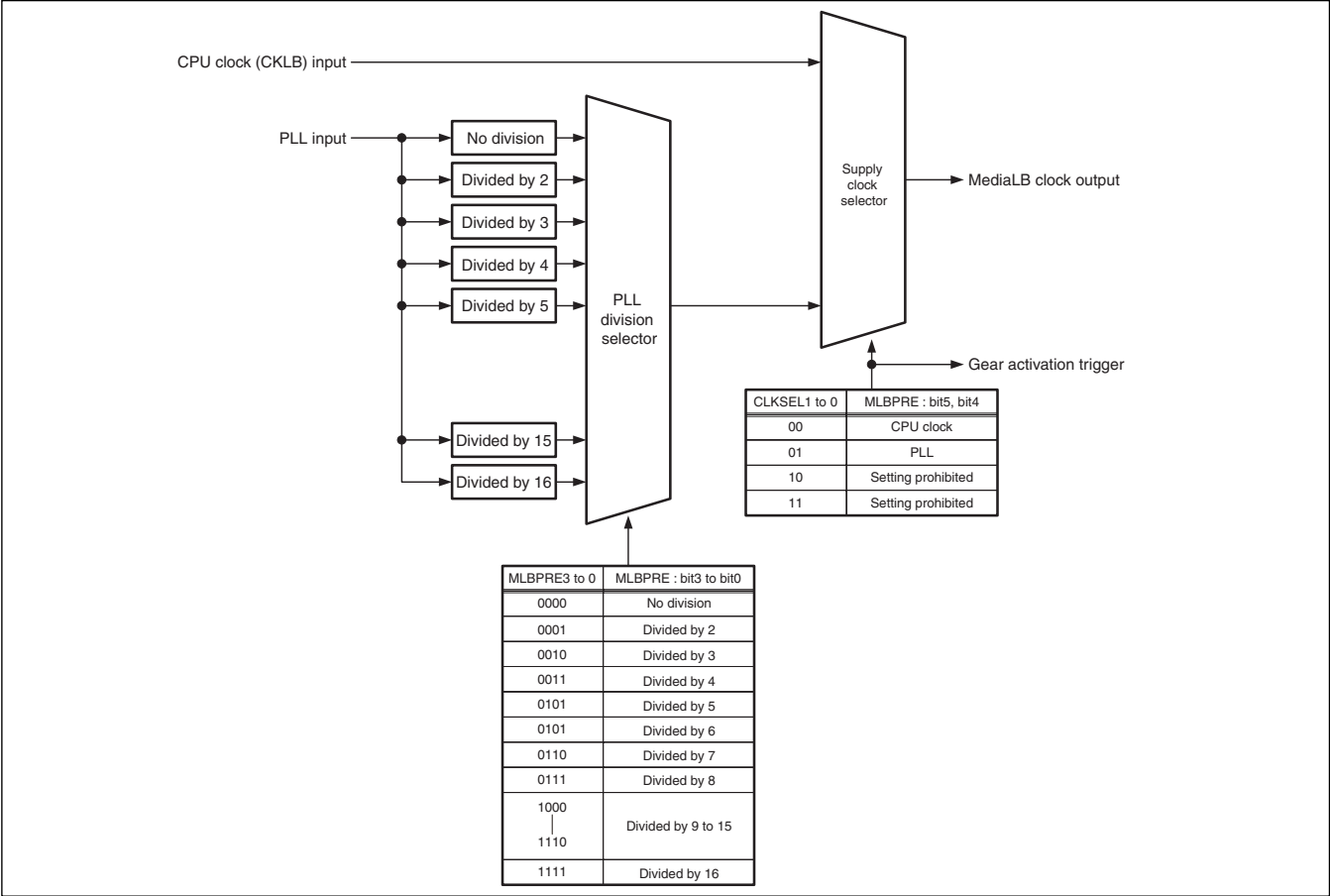
Note:

- Set MPLLDIVM/N/G and MPLLMULG before setting PLLEN to "1".

59.3. Control circuit

59.3.1 Overview

This block consists of a PLL division ratio selector and a CPU clock/PLL clock selector.
Its block diagram is shown below.



Note:

- | | |
|-----------------------------|---|
| CPU clock (CLKB) input: | Core clock input |
| PLL input: | PLL clock input from PLL interface |
| No division/2-16 divisions: | Circuit to divide PLL clock |
| PLL division selector: | Circuit to select divided PLL clock |
| Supply clock selector: | Circuit to select clock supplied to MediaLB macro |
| MediaLB clock output: | Clock supplied to MediaLB macro |
| Gear activation trigger: | Output of clock gear activation trigger |

59.3.2 Registers

■ MLBPRES: MediaLB prescaler control register

The following figure shows the configuration of the MediaLB prescaler control register.

MLBPRES	bit	7	6	5	4	3	2	1	0
Address:0004EB _H		-	-	CLKSEL1	CLKSEL0	MLBPRES3	MLBPRES2	MLBPRES1	MLBPRES0
	R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
	Initial value	-	-	0	0	0	0	0	0

This register divides and selects MediaLB clock.

[bit7, bit6]: Reserved bits

These bits are reserved.

Writing any value does not affect the operation.

[bit5, bit4]: Clock selection bits

These bits select the clock to be supplied to MediaLB macro.

Value	Description
00	CPU clock (CLKB) is supplied as the clock for MediaLB macro.
01	PLL clock is supplied as the clock for MediaLB.
10	Setting prohibited
11	Setting prohibited

At reset, CPU clock is selected.

When PLL is selected, divided PLL clock is supplied to MediaLB macro.

The division ratio is set using bit3 to bit0.

[bit3 to bit0]: PLL clock division ratio setting bits

These bits specify how many divisions should apply to the PLL clock that is output from the PLL interface.

Value	Description
0000 _B	No division
0001 _B	Divided by 2
0010 _B	Setting prohibited
0011 _B	Divided by 4
0100 _B	Setting prohibited
0101 _B	Divided by 6
0110 _B	Setting prohibited
0111 _B	Divided by 8
1000 _B	Setting prohibited

Value	Description
1001 _B	Divided by 10
1010 _B	Setting prohibited
1011 _B	Divided by 12
1100 _B	Setting prohibited
1101 _B	Divided by 14
1110 _B	Setting prohibited
1111 _B	Divided by 16

The frequency of the PLL clock supplied to MediaLB macro is determined by the division ratio of this prescaler and the frequency of the PLL clock output from the PLL interface.

It is prohibited to change this, when PLL is selected as the clock for MediaLB macro (bit5,bit4=01).

Although division by an odd number can also be set, the duty ratio will not be 50:50; therefore, this setting is prohibited.

Notes:

To switch to the PLL clock, the PLL interface must be set to receive the supply of the PLL clock before register setup.

Wait for the stabilization of PLL oscillation, set a division ratio, and then switch the clock.

To switch back to the source clock, switch to the clock, and then stop the PLL clock.

If clock gear operation is selected in the PLL interface, a gear activation trigger will be generated after clock switching, to start the gear operation. Therefore, wait for the completion of the gear operation.

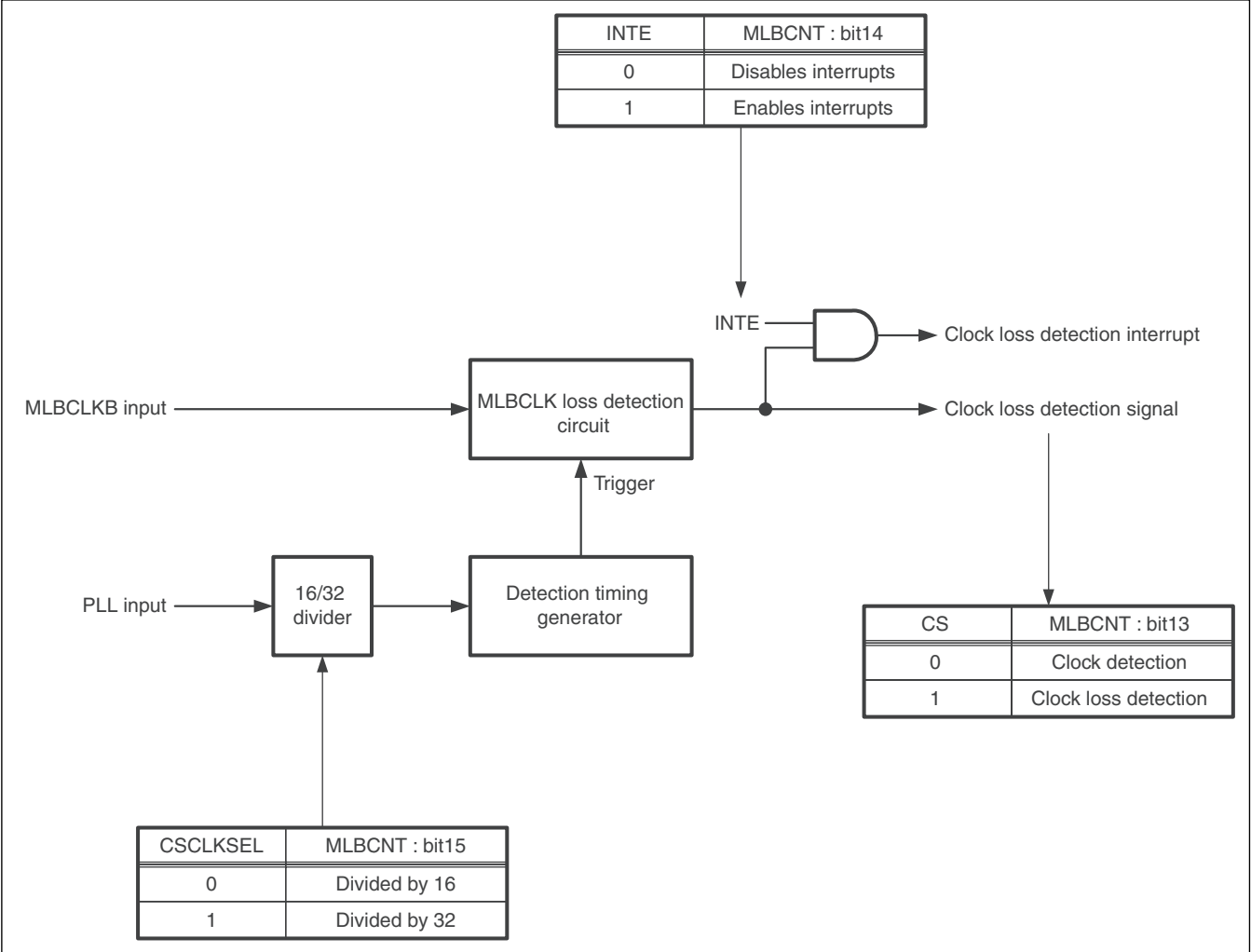
The completion of the gear operation must be detected by either polling the MPLLCTRL register or using an interrupt.

Once the gear operation is completed, either use MediaLB or stop the PLL clock.

59.4. MLBCLK loss detection circuit

59.4.1 Overview

This block is used to detect that MLBCLK has stopped (MLBCLK loss).
The block consists of a MLBCLK counter, a detection timing generator, and a count value checker.
Its block diagram is shown below.



Note:

- MLBCLK input: MLBCLK input from pin

PLL input: PLL clock input from PLL interface

Clock loss detection signal: Signal which indicates that the stop of MLBCLK has been detected

Clock loss detection interrupt: MLBCLK loss detection interrupt

MLBCLK loss detection circuit: Circuit used to detect that MLBCLK has stopped

Detection timing generator: Circuit that generates triggers to detect clock losses

16/32 divider: Divider used to generate detection timings

Detection is performed in a certain cycle.

This detection cycle is generated by the detection timing generator.

The detection cycle is determined by the frequency of PLL input and the 16/32 divider of the above mentioned detection timing generator.

The MLBCLK loss detection circuit detects whether MLBCLK is oscillating by receiving a trigger from the detection timing generator.

Whether or not MLBCLK is stopped is determined by its frequency.

It is determined as "loss", when the frequency of MLBCLK goes below a certain threshold.

The threshold is uniquely defined according to the detection cycle.

The calculation formula used is as follows.

$$\text{CSCLKSEL}=0 : \text{MLBCLK loss detection frequency threshold} = \frac{\text{PLL input frequency}}{8}$$

$$\text{CSCLKSEL}=1 : \text{MLBCLK loss detection frequency threshold} = \frac{\text{PLL input frequency}}{16}$$

An interrupt can be generated when MLBCLK loss is detected.

An interrupt can be enabled or masked using the INTE bit in the MLBCNT register.

This interrupt will be placed on MediaLB interrupt.

The detection cycle used is as follows.

$$\text{CSCLKSEL}=0 : \text{MLBCLK loss detection cycle} = \frac{4}{\text{PLL input frequency}}$$

$$\text{CSCLKSEL}=1 : \text{MLBCLK loss detection cycle} = \frac{16}{\text{PLL input frequency}}$$

The above cycles are used to detect clock loss and recovery.

59.4.2 Registers

■ MLBCNT: MediaLB control register

The following figure shows the configuration of the register.

MLBCNT	bit	15	14	13	12	11	10	9	8
Address:0004EA _H		CCLKSEL	INTE	CS	-	-	-	-	PLLEN
	R/W	R/W	R/W	R	-	-	-	-	R/W
Initial value (INITX or WD-reset)		0	0	0	-	-	-	-	0
Initial value (Software reset)		X	X	X	-	-	-	-	X

WD-reset : Watchdog reset
Initialization X: Not initialized

This register is used to set clock loss detection. It is also used as PLL's operation enable register.

[bit15] CCLKSEL: MLBCLK monitoring clock division setting bit

This bit is used to divide the PLL clock from the PLL interface and set the detection cycle.

CCLKSEL	Division ratio
0	Divided by 16
1	Divided by 32

[bit14] INTE: MLBCLK loss detection interrupt enable bit

This bit specifies whether or not to generate an interrupt when MLBCLK loss is detected.

INTE	Interrupt
0	Does not enable interrupts
1	Enables interrupts

[bit13] CS: MLBCLK loss detection bit

This bit indicates that MLBCLK loss has been detected.

CS	Clock loss
0	Does not detect clock losses
1	Detects clock losses

This bit is set to "0", when a clock with a frequency larger than a set MLBCLK loss detection frequency threshold is input.

This bit is set to "1", when a clock with a frequency smaller than a set MLBCLK loss detection frequency threshold is input.

Writing does not affect the operation.

[bit12 to bit9]: Reserved bits

These bits are reserved.

"0" is always read.

Writing does not affect the operation.

[bit8] PLLEN: MediaLB PLL operation enable bit

For details, see the chapter about the PLL interface.

For reference, the following table shows clock loss detection frequency thresholds used in some cases.
For the registers in the PLL interface, such as DIVM/N, see the chapter about the PLL interface.

PLL interface				Clock loss detection circuit	
Source oscillation [MHz]	DIVM	DIVN	PLL output	CSCLKSEL	Clock loss detection frequency threshold [MHz]
4	2	13	52	0	6.5
4	2	17	68	0	8.5
4	2	21	84	0	10.5
4	2	25	100	1	6.25
4	2	29	116	1	7.25
4	3	13	52	0	6.5
4	3	15	60	0	7.5
4	3	17	68	0	8.5
4	3	19	76	0	9.5

59.5. Interrupt control circuit

59.5.1 Interrupts

For interrupts of each macro, see the chapter about MediaLB.

MediaLB, the FIFO buffer, and I²S operate with a clock that is different from the CPU clock.

Interrupts generated from MediaLB, the FIFO buffer, and I²S are synchronized with the CPU clock before reaching the CPU. The synchronization requires 2 cycles of the CPU clock.

If an interrupt source is erased in a subroutine, for example, it must be synchronized with the clock of MediaLB by reading from the MBSTNC register before recovery.

There are a range of MediaLB-related interrupts, such as MediaLB/FIFO buffer/I²S. As most of them are placed on a single interrupt, an interrupt read register is provided to make it as easy as possible to determine which interrupt is being generated.

59.5.2 Registers

■ MLBINTR: MediaLB interrupt source read register

The following figure shows the configuration of the interrupt source read register.

MLBINTR	bit	31	30	29	28	27	26	25	24
Address:006600 _H		-	CLKSTP	MLBC	MLBS	BUF	-	I2SE9	O2SE8
R/W		R	R	R	R	R	-	R	R
Initial value		0	0	0	0	0	-	0	0
	bit	23	22	21	20	19	18	17	16
Address:006601 _H		I2SE7	I2SE6	I2SE5	I2SE4	I2SE3	I2SE2	I2SE1	I2SE0
R/W		R	R	R	R	R	R	R	R
Initial value		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Address:006602 _H		-	-	-	-	-	-	I2SLR9	I2SLR8
R/W		-	-	-	-	-	-	R/W	R/W
Initial value		-	-	-	-	-	-	0	0
	bit	7	6	5	4	3	2	1	0
Address:006603 _H		I2SLR7	I2SLR6	I2SLR5	I2SLR4	I2SLR3	I2SLR2	I2SLR1	I2SLR0
R/W		R	R	R	R	R	R	R	R
Initial value		0	0	0	0	0	0	0	0

When an interrupt occurs, the corresponding bit is set to "1".

When a MediaLB-related interrupt occurs, this register can be read first to check which interrupt may be occurring.

[bit31 to bit0]: Interrupt source bits

The interrupt sources assigned to these bits are as follows.

Bit	Name	Interrupt source
31	-	None
30	CLKSTP	MLBCLK loss detection interrupt
29	MLBC	MediaLB channel interrupt
28	MLBS	MediaLB system interrupt
27	BUF	FIFO buffer interrupt
26	-	None
25	I2SE9	I ² S Channel 9 error interrupt
24	I2SE8	I ² S Channel 8 error interrupt
23	I2SE7	I ² S Channel 7 error interrupt
22	I2SE6	I ² S Channel 6 error interrupt
21	I2SE5	I ² S Channel 5 error interrupt
20	I2SE4	I ² S Channel 4 error interrupt
19	I2SE3	I ² S Channel 3 error interrupt
18	I2SE2	I ² S Channel 2 error interrupt
17	I2SE1	I ² S Channel 1 error interrupt
16	I2SE0	I ² S Channel 0 error interrupt
15 to 10	-	None
9	I2SLR9	I ² S Channel 9 right channel interrupt or left channel interrupt
8	I2SLR8	I ² S Channel 8 right channel interrupt or left channel interrupt
7	I2SLR7	I ² S Channel 7 right channel interrupt or left channel interrupt
6	I2SLR6	I ² S Channel 6 right channel interrupt or left channel interrupt
5	I2SLR5	I ² S Channel 5 right channel interrupt or left channel interrupt
4	I2SLR4	I ² S Channel 4 right channel interrupt or left channel interrupt
3	I2SLR3	I ² S Channel 3 right channel interrupt or left channel interrupt
2	I2SLR2	I ² S Channel 2 right channel interrupt or left channel interrupt
1	I2SLR1	I ² S Channel 1 right channel interrupt or left channel interrupt
0	I2SLR0	I ² S Channel 0 right channel interrupt or left channel interrupt

59.5.3 DMAC activation

DMA hardware transfer is available for I²S.

When DMA transfer is performed between MediaLB and the FIFO buffer, only software activation is available.

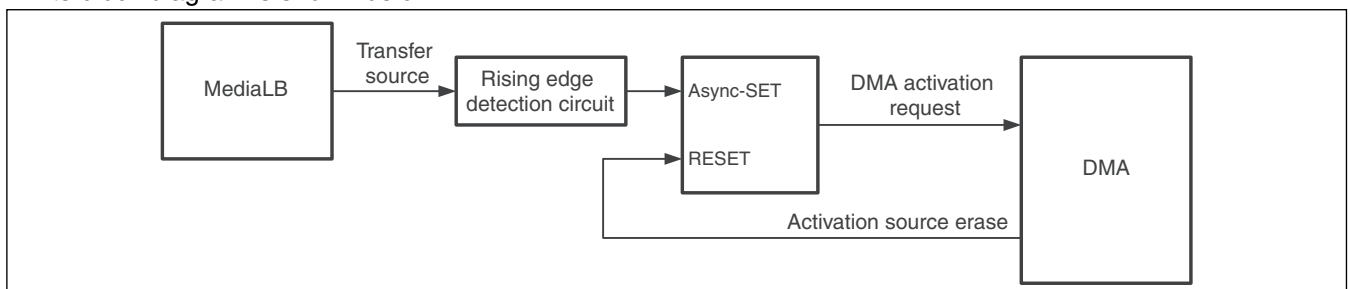
I²S operates with a clock that is different from the CPU clock.

Therefore, DMA transfer sources reach DMAC via a junction circuit.

In this case, the junction circuit detects the rising edge of a transfer source (interrupt) and transmits the source to DMAC.

In other words, this can be seen as there is an external interrupt circuit between the FIFO buffer/I²S and DMAC.

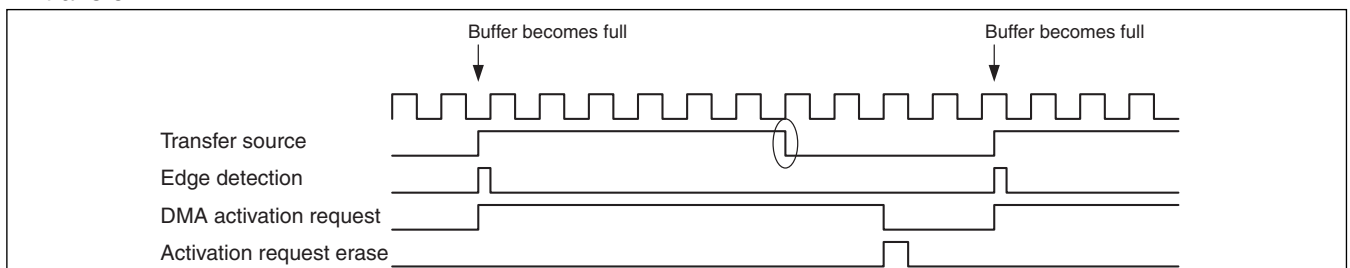
Its block diagram is shown below.



If a DMA transfer is not completed before the transfer source falls, the transfer source continues to occur. Consequently, the next transfer source cannot be accepted.

An example of this is shown on the following page.

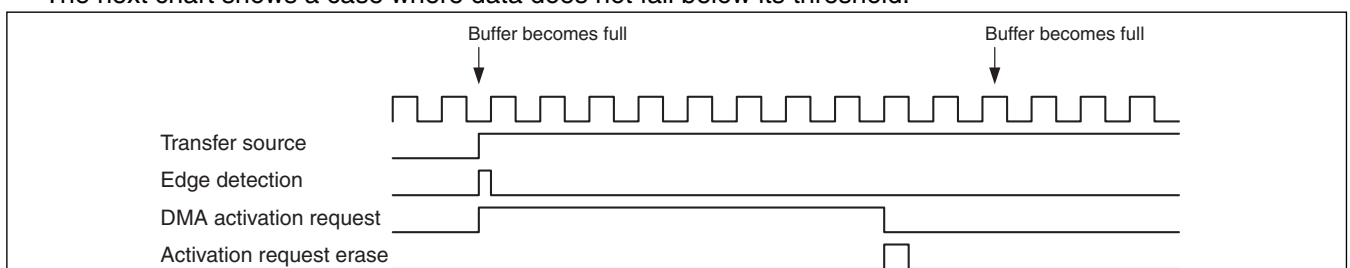
The following figure shows a timing chart for when data in the buffer falls below its threshold due to DMA transfer.



The buffer becomes full twice. As data declines below its threshold in the first DMA transfer, the transfer source falls.

When the buffer becomes full for the second time, edge detection occurs again. Therefore, the second DMA activation also occurs.

The next chart shows a case where data does not fall below its threshold.



Since the transfer source does not fall, edge detection does not occur and the second activation request is not generated.

To prevent this, set the threshold and the number of transfers appropriately.

59.6. Bus interface

MediaLB/the FIFO buffer/I²S operate with a clock that is different from the CPU clock. Whenever the CPU and DMA access registers, the different clocks are always synchronized.

The time required for synchronization varies depending on the speeds of the CPU clock and the MediaLB operation clock. The transfer speed increases according to the speeds of both clocks.

If the bus bandwidth of MediaLB is limited, it can be extended by increasing the operation clock of MediaLB to its maximum speed (112MHz). The power consumption however increases.

Chapter 60 MediaLB (Media Local Bus interface)

This chapter describes the MediaLB features.

60.1. Overview of the MediaLB

The MediaLB comprises a MediaLB functional block, HB decoder, FIFO buffer, HBI bus and PBI bus. It supports transfer rates of 256FS and 512FS. A 2K word RAM for the local channel buffer and another 2K word RAM for the FIFO buffer are provided.

■ Overview

The MediaLB is an interface that implements a local bus specification MediaLB (Media Local Bus) for connecting IC devices with a MOST (Media Oriented System Transport, which is a multimedia-type in-vehicle LAN standard) network.

A MOST network is formed by connecting several MOST network interface controllers. The MediaLB implements a local serial bus for connecting a plurality of ICs to MOST network interface controllers.

This LSI has an I²S, enabling data transfer between the I²S and MediaLB.

Also, data can be exchanged with the MediaLB from CPU via an F-bus.

Its features are shown below.

		Function
1	Data transfer	Supports sync data transfer (Sync), async data transfer (Async), and control data transfer (Control).
2	Channels	15 channels
3	Transfer rate	Supports 256Fs and 512Fs.
4	Operation mode	Supports DMA mode (ping-pong buffering and circular buffering) and IO mode.
5	RAM	Local channel buffer : 2048 words ¥ 32 bits FIFO buffer : 2048 words ¥ 32 bits

For details on the MediaLB bus protocol, see "Media Local Bus Specification 3.0".

■ Term descriptions

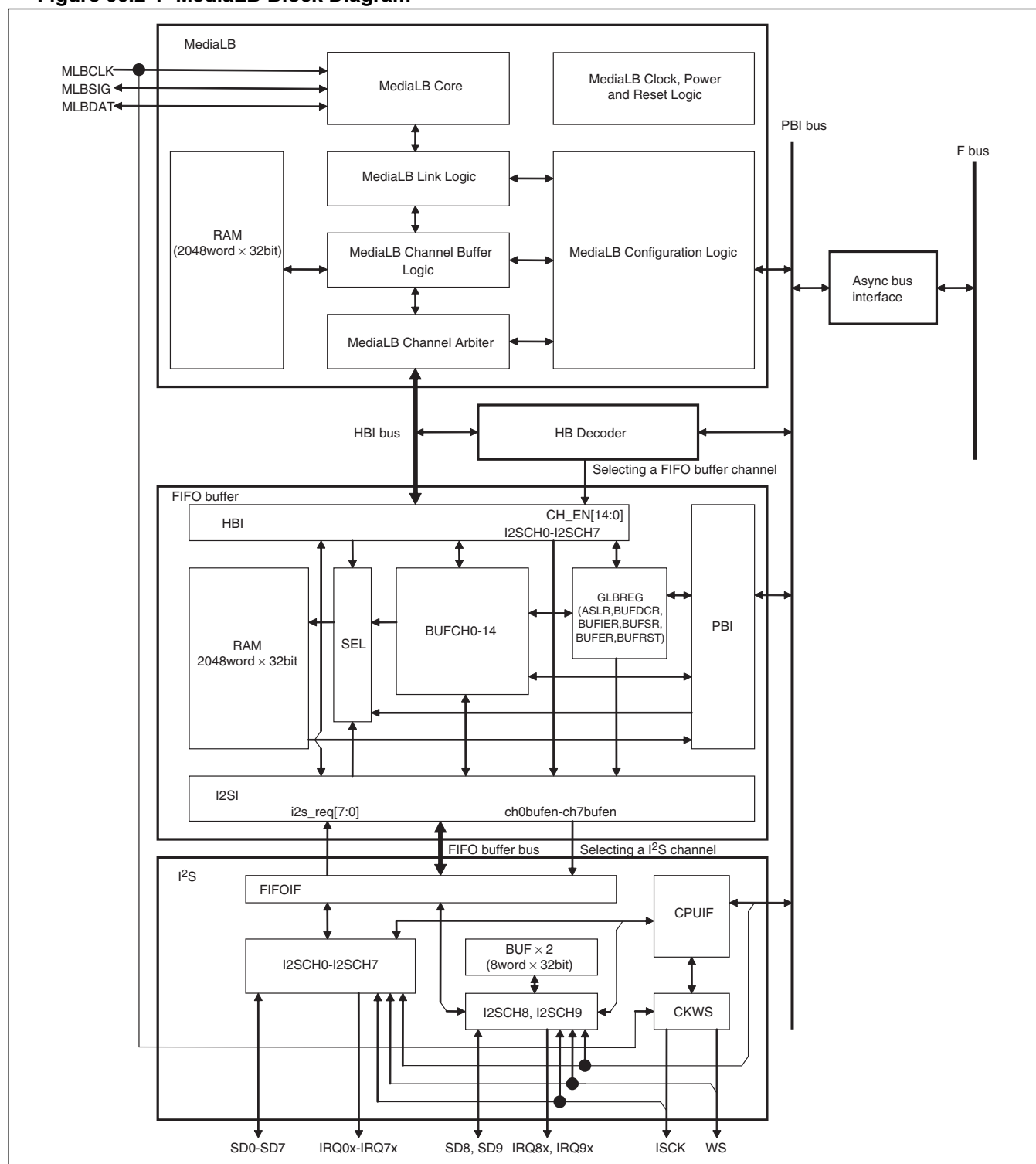
Table 60.1-1 shows the descriptions of the terms used in this chapter.

Table 60.1-1 Term Descriptions

Term	Description
MediaLB	The function block that implements MediaLB (Media Local Bus), which is a local bus specification, for connecting IC devices with a MOST (Media Oriented System Transport, which is a multimedia-type in-vehicle LAN standard) network.
Sync data	Data of stream signals, such as audio data.
Async data	Packet data.
Control data	Data of control signals.
Local channel buffer	The data storage area in the MediaLB.
FIFO buffer	The hardware block, provided between the MediaLB and I ² S, consisting of a data area (FIFO channel buffer) for temporarily storing data and a logic section for controlling the area.
FIFO channel buffer	The data area, provided inside the FIFO buffer, for temporarily storing data.
HBI bus	The local bus for connecting the MediaLB and FIFO buffer.
IO mode	The mode capable of exchanging transmission/reception data with the MediaLB from CPU.
DMA mode	The mode in which the MediaLB operates as the bus master of the HBI bus. Transmission/reception data is exchanged between the MediaLB and FIFO channel buffer via the HBI bus.
Current buffer	The data area on the HBI bus that is read/written in DMA mode.
Current buffer address	Indicates the address area that is output on the HBI bus in DMA mode. Indicated in the channel n current buffer configuration register (CCBCRn).
Next buffer	The data area that is processed next on the HBI bus in DMA mode after processing the current buffer area.
Next buffer address	Indicates the address area that is output next on the HBI bus in DMA mode after processing the current buffer area. Set in the channel n next buffer configuration register (CNBCRn).
Previous buffer	The data area processed before the current buffer in DMA mode.
Previous buffer address	The address area processed before the current buffer in DMA mode.
Quadlet	Indicates a 4-byte unit.
Ping-pong buffering	One of two data transfer methods in DMA mode. If the current buffer is filled, an interrupt occurs when moving to the next buffer and so the next address to be used next can be set again by software. Therefore, it is possible to transmit/receive data while changing the transfer destination data storage area one after another.
Circular buffering	One of two data transfer methods in DMA mode. Performs data transfers repeatedly for the data storage area indicated by the current buffer. No interrupt occurs at the completion of a transfer.

Displays the entire MediaLB block diagram including I²S.

Figure 60.2-1 MediaLB Block Diagram



■ Descriptions of the Block Diagram

● MediaLB

The function block of MediaLB.

A RAM (2048 words 32 bits) for local channel buffer is embedded internally. Transmission/Reception is enabled in sync, async and control channels, and it supports 15 logical channels.

- MediaLB Clock, Power and Reset Logic

The MediaLB Clock, Power and Reset Logic controls the clock, power and reset of the MediaLB.

- MediaLB Link Logic

The MediaLB Link Logic supports the following functions of the MediaLB interface link layer.

- Protocol check for sync channels, async channels and control channels
- Break processing in transmission/reception
- Generation of reception responses
- Generation of transmission commands
- Detection of lock/unlock of the MediaLB bus
- Comparison and detection of logical channels

- MediaLB Configuration Logic

The MediaLB Configuration Logic controls the registers in MediaLB. These registers are accessed via the PBI bus.

- MediaLB Channel Buffer Logic

The MediaLB Channel Buffer Logic controls the interface with RAM (2048 words × 32 bits). MediaLB Channel Buffer Logic also supports the following functions.

- Loop back mode between the logical channel 0 (reception) and logical channel 1 (transmission)
- Buffering of logical channel data for solving bus latency

- MediaLB Channel Arbiter

The MediaLB Channel Arbiter controls the HBI bus as its bus master in DMA mode and responds a request from the MediaLB bus. MediaLB Channel Arbiter also supports the following functions.

- Control of the FIFO buffer bus as its bus master
- Determination of the priority order of channels in DMA mode
- Selection of data and control lines between the HBI interface and MediaLB logical channel
- Integration of channel interrupts

- MediaLB Core

The MediaLB Core supports the physical layer of the MediaLB interface. This physical layer converts serial data to parallel data and vice versa. It also synchronizes data in the MediaLB frame.

● HB Decoder

HB Decoder compares the address on the HBI bus and the data set in each FIFO buffer channel n address range register, and if the address exists within the range set by the register n, it selects the FIFO buffer channel n and exchanges data between the MediaLB and FIFO buffer via the HBI bus. If the address on the HBI bus is out of the range set by the address range register, the HB Decoder returns an error signal to the MediaLB. When the MediaLB receives the error, it sets STS[5] in the channel n status configuration register to "1".

This block is used only when the MediaLB is in DMA mode.

● FIFO buffer

The FIFO buffer is used as a transmission/reception buffer of the MediaLB and a transmission/reception buffer of ch.0-ch.7 in the I²S. A RAM with 2048 words × 32 bits is embedded internally.

Ch.0 in the FIFO buffer is connected with ch.0 in the I²S, and ch.7 in the FIFO buffer is connected with ch.7 in the I²S. For MediaLB, a channel of the MediaLB and a channel of the FIFO buffer can be connected arbitrarily by setting the FIFO buffer channel n address range register of the HB Decoder.

- PBI

PBI bus interface block. It controls registers and the RAM for the access from the PBI bus.

- HBI

HBI bus interface block. It controls registers and the RAM for the access from the HBI bus.

- I²SI

The interface block with the I²S. It controls the FIFO buffer bus as its bus master. This block access to the FIFO buffer bus when receiving a data request signal from the I²S.

- GLBREG

The group of registers, including the access select register (ASLR), FIFO buffer direction control register (BUFDCR), FIFO buffer interrupt register (BUFIER), FIFO buffer status register (BUFSR), FIFO buffer error register (BUFER) and FIFO buffer reset register (BUFRST).

- BUFCH0-14

Controls the read pointer, write pointer and the number of valid quadlets for each channel from channel 0 to channel 14. It performs arbitration for each access from the I²S, HBI bus and PBI bus.

- SEL

Selects data written from the I²S, HBI bus and PBI bus.

- RAM

RAM with 2048 words × 32 bits. It temporarily stores data.

● I²S

The function block of I²S. Comprises 10ch (channels 0 to 9), and has a buffer with 8 words × 32 bits for channel 8 and channel 9. It supports both master and slave.

- CPUIF

Has the I²S common control register (I2SCCR) and I²S rate setting register (I2SRSR). The interface with the PBI bus.

- FIFOIF

The interface with the FIFO buffer. If a plurality of I²S channels is selected, it outputs an error to the FIFO buffer to set the bit of the FIFO buffer error register in the FIFO buffer to "1".

- BUF × 2

The buffers for I²S channels 8 and 9. The configuration with 8 words × 32 bits provided for two channels.

- I2SCH0-I2SCH9

The block for converting serial data to parallel data and parallel data to serial data for the I²S channels 0 to 9.

● PBI bus (PBI=Peripheral Bus Interface)

The bus is used to access to registers in each block of MediaLB, HB Decoder, FIFO buffer and I²S.

Also, this bus is used in data transfer with the I²S, FIFO buffer and MediaLB.

● HBI bus (HBI=Host Bus Interface)

The bus is used in data transfer between the MediaLB and FIFO buffer. When the MediaLB is in DMA mode, this bus becomes its bus master and transmission/reception data in the MediaLB frame is exchanged on this bus.

60.3. MediaLB Registers

A list of the MediaLB registers is shown below.

■ List of the MediaLB Registers

Table 60.3-1 List of the MediaLB Registers (1 / 5)

Address	bit31	bit0
006000 _H	Device Control Configuration Register (DCCR)	
006004 _H	System Status Configuration Register (SSCR)	
006008 _H	System Data Configuration Register (SDCR)	
00600C _H	System Mask Configuration Register (SMCR)	
00601C _H	Version Control Configuration Register (VCCR)	
006030 _H	Channel Interrupt Configuration Register (CICR)	
Channel 0 register		
006040 _H	Channel 0 Entry Configuration Register (CECR0)	
006044 _H	Channel 0 Status Configuration Register (CSCR0)	
006048 _H	Channel 0 Current Buffer Configuration Register (CCBCR0)	
00604C _H	Channel 0 Next Buffer Configuration Register (CNBCR0)	
006280 _H	Channel 0 Local Channel Buffer Configuration Register (LCBCR0)	
Channel 1 register		
006050 _H	Channel 1 Entry Configuration Register (CECR1)	
006054 _H	Channel 1 Status Configuration Register (CSCR1)	
006058 _H	Channel 1 Current Buffer Configuration Register (CCBCR1)	
00605C _H	Channel 1 Next Buffer Configuration Register (CNBCR1)	
006284 _H	Channel 1 Local Channel Buffer Configuration Register (LCBCR1)	
Channel 2 register		
006060 _H	Channel 2 Entry Configuration Register (CECR2)	
006064 _H	Channel 2 Status Configuration Register (CSCR2)	
006068 _H	Channel 2 Current Buffer Configuration Register (CCBCR2)	
00606C _H	Channel 2 Next Buffer Configuration Register (CNBCR2)	
006288 _H	Channel 2 Local Channel Buffer Configuration Register (LCBCR2)	
Channel 3 register		
006070 _H	Channel 3 Entry Configuration Register (CECR3)	
006074 _H	Channel 3 Status Configuration Register (CSCR3)	
006078 _H	Channel 3 Current Buffer Configuration Register (CCBCR3)	
00607C _H	Channel 3 Next Buffer Configuration Register (CNBCR3)	
00628C _H	Channel 3 Local Channel Buffer Configuration Register (LCBCR3)	

MB91460 Series**Table 60.3-1 List of the MediaLB Registers (2 / 5)**

Address	bit31	bit0
Channel 4 register		
006080 _H	Channel 4 Entry Configuration Register (CECR4)	
006084 _H	Channel 4 Status Configuration Register (CSCR4)	
006088 _H	Channel 4 Current Buffer Configuration Register (CCBCR4)	
00608C _H	Channel 4 Next Buffer Configuration Register (CNBCR4)	
006290 _H	Channel 4 Local Channel Buffer Configuration Register (LCBCR4)	
Channel 5 register		
006090 _H	Channel 5 Entry Configuration Register (CECR5)	
006094 _H	Channel 5 Status Configuration Register (CSCR5)	
006098 _H	Channel 5 Current Buffer Configuration Register (CCBCR5)	
00609C _H	Channel 5 Next Buffer Configuration Register (CNBCR5)	
006294 _H	Channel 5 Local Channel Buffer Configuration Register (LCBCR5)	
Channel 6 register		
0060A0 _H	Channel 6 Entry Configuration Register (CECR6)	
0060A4 _H	Channel 6 Status Configuration Register (CSCR6)	
0060A8 _H	Channel 6 Current Buffer Configuration Register (CCBCR6)	
0060AC _H	Channel 6 Next Buffer Configuration Register (CNBCR6)	
006298 _H	Channel 6 Local Channel Buffer Configuration Register (LCBCR6)	
Channel 7 Register		
0060B0 _H	Channel 7 Entry Configuration Register (CECR7)	
0060B4 _H	Channel 7 Status Configuration Register (CSCR7)	
0060B8 _H	Channel 7 Current Buffer Configuration Register (CCBCR7)	
0060BC _H	Channel 7 Next Buffer Configuration Register (CNBCR7)	
00629C _H	Channel 7 Local Channel Buffer Configuration Register (LCBCR7)	
Channel 8 Register		
0060C0 _H	Channel 8 Entry Configuration Register (CECR8)	
0060C4 _H	Channel 8 Status Configuration Register (CSCR8)	
0060C8 _H	Channel 8 Current Buffer Configuration Register (CCBCR8)	
0060CC _H	Channel 8 Next Buffer Configuration Register (CNBCR8)	
0062A0 _H	Channel 8 Local Channel Buffer Configuration Register (LCBCR8)	
Channel 9 register		
0060D0 _H	Channel 9 Entry Configuration Register (CECR9)	
0060D4 _H	Channel 9 Status Configuration Register (CSCR9)	
0060D8 _H	Channel 9 Current Buffer Configuration Register (CCBCR9)	
0060DC _H	Channel 9 Next Buffer Configuration Register (CNBCR9)	

Table 60.3-1 List of the MediaLB Registers (3 / 5)

Address	bit31	bit0
0062A4 _H	Channel 9 Local Channel Buffer Configuration Register (LCBCR9)	
Channel 10 Register		
0060E0 _H	Channel 10 Entry Configuration Register (CECR10)	
0060E4 _H	Channel 10 Status Configuration Register (CSCR10)	
0060E8 _H	Channel 10 Current Buffer Configuration Register (CCBCR10)	
0060EC _H	Channel 10 Next Buffer Configuration Register (CNBCR10)	
0062A8 _H	Channel 10 Local Channel Buffer Configuration Register (LCBCR10)	
Channel 11 Register		
0060F0 _H	Channel 11 Entry Configuration Register (CECR11)	
0060F4 _H	Channel 11 Status Configuration Register (CSCR11)	
0060F8 _H	Channel 11 Current Buffer Configuration Register (CCBCR11)	
0060FC _H	Channel 11 Next Buffer Configuration Register (CNBCR11)	
0062AC _H	Channel 11 Local Channel Buffer Configuration Register (LCBCR11)	
Channel 12 Register		
006100 _H	Channel 12 Entry Configuration Register (CECR12)	
006104 _H	Channel 12 Status Configuration Register (CSCR12)	
006108 _H	Channel 12 Current Buffer Configuration Register (CCBCR12)	
00610C _H	Channel 12 Next Buffer Configuration Register (CNBCR12)	
0062B0 _H	Channel 12 Local Channel Buffer Configuration Register (LCBCR12)	
Channel 13 Register		
006110 _H	Channel 13 Entry Configuration Register (CECR13)	
006114 _H	Channel 13 Status Configuration Register (CSCR13)	
006118 _H	Channel 13 Current Buffer Configuration Register (CCBCR13)	
00611C _H	Channel 13 Next Buffer Configuration Register (CNBCR13)	
0062B4 _H	Channel 13 Local Channel Buffer Configuration Register (LCBCR13)	
Channel 14 Register		
006120 _H	Channel 14 Entry Configuration Register (CECR14)	
006124 _H	Channel 14 Status Configuration Register (CSCR14)	
006128 _H	Channel 14 Current Buffer Configuration Register (CCBCR14)	
00612C _H	Channel 14 Next Buffer Configuration Register (CNBCR14)	
0062B8 _H	Channel 14 Local Channel Buffer Configuration Register (LCBCR14)	
Bus Interface Register		
006410 _H	Buffer Channel 0 Address Range Register (BUFAR0)	
006414 _H	Buffer Channel 1 Address Range Register (BUFAR1)	
006418 _H	Buffer Channel 2 Address Range Register (BUFAR2)	

Table 60.3-1 List of the MediaLB Registers (4 / 5)

Address	bit31	bit0
00641C _H	Buffer Channel 3 Address Range Register (BUFAR3)	
006420 _H	Buffer Channel 4 Address Range Register (BUFAR4)	
006424 _H	Buffer Channel 5 Address Range Register (BUFAR5)	
006428 _H	Buffer Channel 6 Address Range Register (BUFAR6)	
00642C _H	Buffer Channel 7 Address Range Register (BUFAR7)	
006430 _H	Media Stereo Data Setting Register (MSTD)	
006434 _H	Buffer Channel 8 Address Range Register (BUFAR8)	
006438 _H	Buffer Channel 9 Address Range Register (BUFAR9)	
00643C _H	Buffer Channel 10 Address Range Register (BUFAR10)	
006440 _H	Buffer Channel 11 Address Range Register (BUFAR11)	
006444 _H	Buffer Channel 12 Address Range Register (BUFAR12)	
006448 _H	Buffer Channel 13 Address Range Register (BUFAR13)	
00644C _H	Buffer Channel 14 Address Range Register (BUFAR14)	
FIFO buffer		
0064A0 _H	Access Select Register (ASLR)	
0064A4 _H	FIFO Buffer Direction Control Register (BUFDCR)	
0064A8 _H	FIFO Buffer Interrupt Enable Register (BUFIER)	
0064AC _H	FIFO Buffer Status Register (BUFSR)	
0064B0 _H	FIFO Buffer Error Register (BUFER)	
0064B4 _H	FIFO Buffer Reset Register (BUFRST)	
0064B8 _H -0064BF _H	Reserved	
0064C0 _H	FIFO Buffer Count Register 0 (BUFCT0)	
0064C4 _H	FIFO Buffer Count Register 1 (BUFCT1)	
0064C8 _H	FIFO Buffer Count Register 2 (BUFCT2)	
0064CC _H	FIFO Buffer Count Register 3 (BUFCT3)	
0064D0 _H	FIFO Buffer Count Register 4 (BUFCT4)	
0064D4 _H	FIFO Buffer Count Register 5 (BUFCT5)	
0064D8 _H	FIFO Buffer Count Register 6 (BUFCT6)	
0064DC _H	FIFO Buffer Count Register 7 (BUFCT7)	
0064E0 _H -0064FF _H	Reserved	
006500 _H	FIFO Buffer 0 Control Register (BUF0CR)	
006504 _H	FIFO Buffer 1 Control Register (BUF1CR)	
006508 _H	FIFO Buffer 2 Control Register (BUF2CR)	
00650C _H	FIFO Buffer 3 Control Register (BUF3CR)	
006510 _H	FIFO Buffer 4 Control Register (BUF4CR)	

Table 60.3-1 List of the MediaLB Registers (5 / 5)

Address	bit31	bit0
006514 _H	FIFO Buffer 5 Control Register (BUF5CR)	
006518 _H	FIFO Buffer 6 Control Register (BUF6CR)	
00651C _H	FIFO Buffer 7 Control Register (BUF7CR)	
006520 _H	FIFO Buffer 8 Control Register (BUF8CR)	
006524 _H	FIFO Buffer 9 Control Register (BUF9CR)	
006528 _H	FIFO Buffer 10 Control Register (BUF10CR)	
00652C _H	FIFO Buffer 11 Control Register (BUF11CR)	
006530 _H	FIFO Buffer 12 Control Register (BUF12CR)	
006534 _H	FIFO Buffer 13 Control Register (BUF13CR)	
006538 _H	FIFO Buffer 14 Control Register (BUF14CR)	
00653C _H -00657F _H	Reserved	
006580 _H	FIFO Buffer 0 Data Register (BUF0DTR)	
006584 _H	FIFO Buffer 1 Data Register (BUF1DTR)	
006588 _H	FIFO Buffer 2 Data Register (BUF2DTR)	
00658C _H	FIFO Buffer 3 Data Register (BUF3DTR)	
006590 _H	FIFO Buffer 4 Data Register (BUF4DTR)	
006594 _H	FIFO Buffer 5 Data Register (BUF5DTR)	
006598 _H	FIFO Buffer 6 Data Register (BUF6DTR)	
00659C _H	FIFO Buffer 7 Data Register (BUF7DTR)	
0065A0 _H	FIFO Buffer 8 Data Register (BUF8DTR)	
0065A4 _H	FIFO Buffer 9 Data Register (BUF9DTR)	
0065A8 _H	FIFO Buffer 10 Data Register (BUF10DTR)	
0065A4 _H	FIFO Buffer 11 Data Register (BUF11DTR)	
0065A8 _H	FIFO Buffer 12 Data Register (BUF12DTR)	
0065A4 _H	FIFO Buffer 13 Data Register (BUF13DTR)	
0065A8 _H	FIFO Buffer 14 Data Register (BUF14DTR)	
0065AC _H -0065FF _H	Reserved	

MB91460 Series**MediaLB bit allocations****Table 60.3-2 MediaLB Bit Allocation**

• Upper 16 bits

		Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
DDCR		MDE	LBM	MCS1	MCS0	M5PS	MLK	MLE	MHRE	MRS							
SSCR																	
SDCR		MSD31	MSD30	MSD29	MSD28	MSD27	MSD26	MSD25	MSD24	MSD23	MSD22	MSD21	MSD20	MSD19	MSD18	MSD17	MSD16
SMCR																	
VCCR										CSC7	CSC6	CSC5	CSC4	CSC3	CSC2	CSC1	CSC0
CICR																	
CECRn		CE	TR	CT1	CT0	CNTE	MDS1	MDS0	-	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
CSCRn		BM	BF											IVB1	IVB0	GB	RDY
CCBCRn	IO	RDB31	RDB30	RDB29	RDB28	RDB27	RDB26	RDB25	RDB24	RDB23	RDB22	RDB21	RDB20	RDB19	RDB18	RDB17	RDB16
	DMA	BCA15	BCA14	BCA13	BCA12	BCA11	BCA10	BCA9	BCA8	BCA7	BCA6	BCA5	BCA4	BCA3	BCA2	BCA1	BCA0
CNBCRn	IO	TDB31	TDB30	TDB29	TDB28	TDB27	TDB26	TDB25	TDB24	TDB23	TDB22	TDB21	TDB20	TDB19	TDB18	TDB17	TDB16
	DMA	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
LCBCRn		TH9	TH8	TH7	TH6	TH5	TH4	TH3	TH2	TH1	TH0	BD8	BD7	BD6	BD5	BD4	BD3
BUFARn		ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8	ST7	ST6	ST5	ST4	ST3	ST2	-	-
MSTD		-	MSTD14	MSTD13	MSTD12	MSTD11	MSTD10	MSTD9	MSTD8	MSTD7	MSTD6	MSTD5	MSTD4	MSTD3	MSTD2	MSTD1	MSTD0
ASLR		AS7[1:0]		AS6[1:0]		AS5[1:0]		AS4[1:0]		AS3[1:0]		AS2[1:0]		AS1[1:0]		AS0[1:0]	
BUFDCR		-	BRD14	BRD13	BRD12	BRD11	BRD10	BRD9	BRD8	BRD7	BRD6	BRD5	BRD4	BRD3	BRD2	BRD1	BRD0
BUFIER		-	BIRE14	BIRE13	BIRE12	BIRE11	BIRE10	BIRE9	BIRE8	BIRE7	BIRE6	BIRE5	BIRE4	BIRE3	BIRE2	BIRE1	BIRE0
BUFSR		-	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
BUFER		-	BER14	BER13	BER12	BER11	BER10	BER9	BER8	BER7	BER6	BER5	BER4	BER3	BER2	BER1	BER0
BUFRST		-	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
BUFCTm						BCT2m[11:8]				BCT2m[7:0]							
BUFnCR		BTH[9:2]								BTH[1:0]			BDP[8:3]				
BUFnDTR		BUF[31:24]								BUF[23:16]							

• Lower 16 bits

		Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08	Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
DDCR		-								MDA8	MDA7	MDA6	MDA5	MDA4	NDA3	MDA2	MDA1
SSCR		-								SSRE	SDMU	SDML	SDSC	SDCS	SDNU	SDNL	SDR
SDCR		MSD15	MSD14	MSD13	MSD12	MSD11	MSD10	MSD9	MSD8	MSD7	MSD6	MSD5	MSD4	MSD3	MSD2	MSD1	MSD0
SMCR		-								-	SMMU	SMML	SMSC	SMCS	SMNU	SMNL	SMR
VCCR		MMA7	MMA6	MMA5	MMA4	MMA3	MMA2	MMA1	MMA0	MMI7	MMI6	MMI5	MMI4	MMI3	MMI2	MMI1	MMI0
CICR		-	CNSU14	CNSU13	CNSU12	CNSU11	CNSU10	CNSU9	CNSU8	CNSU7	CNSU6	CNSU5	CNSU4	CNSU3	CNSU2	CNSU1	CNSU0
CECRn		PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	CA8	CA7	CA6	CA5	CA4	CA3	CA2	CA1
CSCRn		STS15	STS14	STS13	STS12	STS11	STS10	STS9	STS8	STS7	STS6	STS5	STS4	STS3	STS2	STS1	STS0
CCBCRn	IO	RDB15	RDB14	RDB13	RDB12	RDB11	RDB10	RDB9	RDB8	RDB7	RDB6	RDB5	RDB4	RDB3	RDB2	RDB1	RDB0
	DMA	BFA15	BFA14	BFA13	BFA12	BFA11	BFA10	BFA9	BFA8	BFA7	BFA6	BFA5	BFA4	BFA3	BFA2	BFA1	BFA0
CNBCRn	IO	TDB15	TDB14	TDB13	TDB12	TDB11	TDB10	TDB9	TDB8	TDB7	TDB6	TDB5	TDB4	TDB3	TDB2	TDB1	TDB0
	DMA	BEA15	BEA14	BEA13	BEA12	BEA11	BEA10	BEA9	BEA8	BEA7	BEA6	BEA5	BEA4	BEA3	BEA2	BEA1	BEA0
LCBCRn		BD2	BD1	BD0	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
BUFARn		EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8	EA7	EA6	EA5	EA4	EA3	EA2	-	IFEN
MSTD		-								-							
ASLR		QD7[1:0]		QD6[1:0]		QD5[1:0]		QD4[1:0]		QD3[1:0]		QD2[1:0]		QD1[1:0]		QD0[1:0]	
BUFDCR		I2SRQE	-							-							
BUFIER		-								-							
BUFSR		-								-							
BUFER		-								-							
BUFRST		-								-							
BUFCTm+1		-				BCTm+1[11:8]				BCTm+1[7:0]							
BUFnCR		BDP[2:0]			-				BSA[8]		BSA[7:0]						
BUFnDTR		BUF[15:8]								BUF[7:0]							

60.3.1 Device Control configuration Register (DCCR)

Device Control Configuration Register (DCCR) is a register to set the permission, clock rate, pin mode, lock status, software reset, and device address of MediaLB.

■ **Device Control Configuration Register (DCCR)**

Figure 60.3-1 displays the bit configuration of the Device Control Configuration Register (DCCR), and Table 60.3-3 displays the function of each bit.

Figure 60.3-1 Bit Configuration of the Device Control Configuration Register (DCCR)

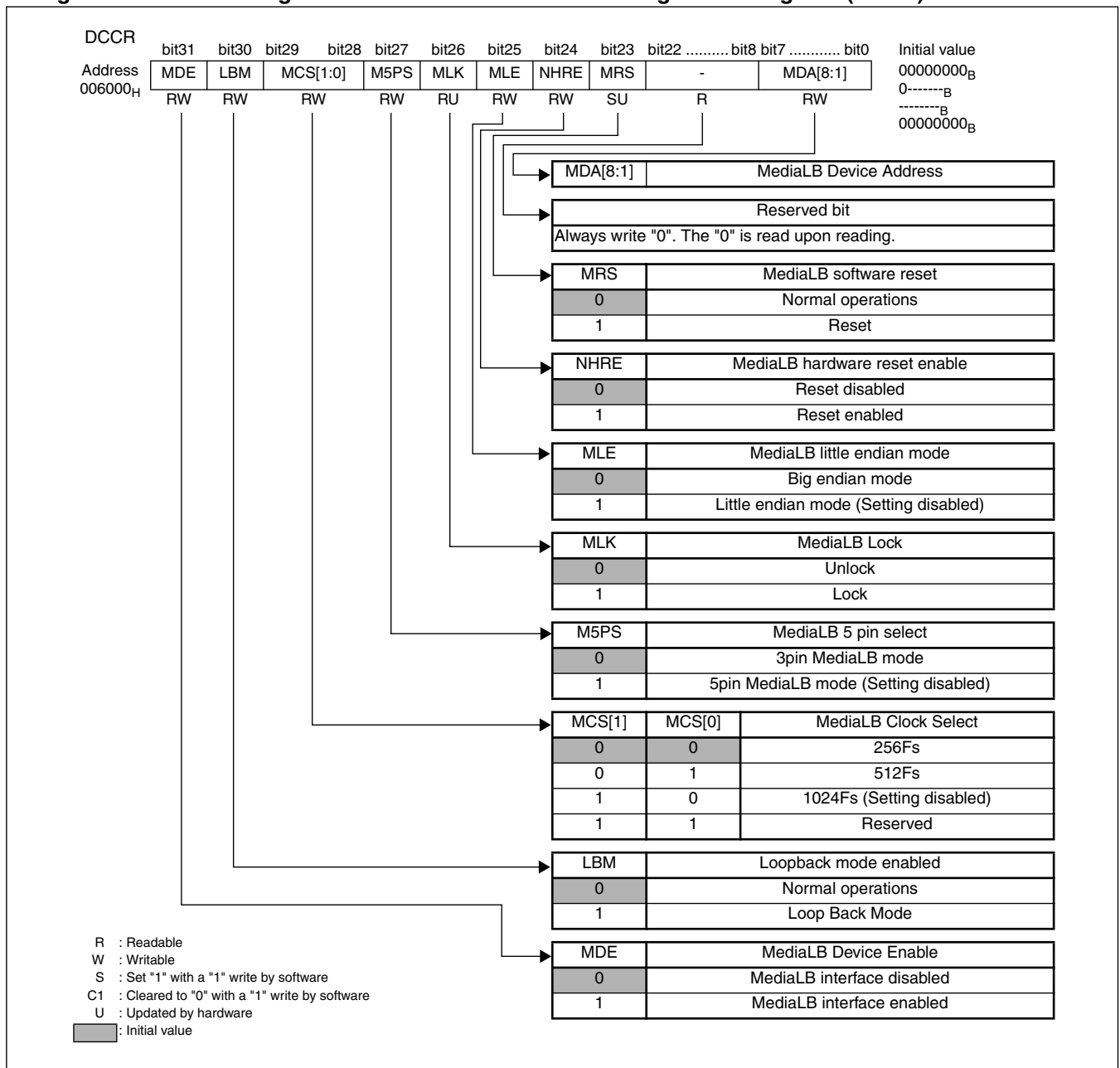


Table 60.3-3 Functional Descriptions on the Device Control Configuration Register (DCCR) Bits (1 / 2)

Bit Name		Function
bit31	MDE: MediaLB Device Enable	Enables the operation of the MediaLB interface. Set to "0": Prohibits the operation of the MediaLB interface. Set to "1": Enables the operation of the MediaLB interface.
bit30	LBM: Loop Back Mode Enable	Sets whether the loopback test for the MediaLB bus between logical channel 0 (reception setting) and logical channel 1 (transmission setting) is enabled or disabled. Set to "0": Normal operation Set to "1": Loopback test mode

Table 60.3-3 Functional Descriptions on the Device Control Configuration Register (DCCR) Bits (2 / 2)

Bit Name		Function
bit29, bit28	MCS[1:0]: MediaLB Clock Select	Sets the MediaLB transfer route. 1024Fs is not supported. Set to "00": 256Fs: Supports 8 quadlets (1 quadlet=4 bytes) per frame. Set to "01": 512Fs: Supports 16 quadlets per frame. Set to "10": 1024Fs: Supports 32 quadlets per frame. (It is prohibited to set this.) <Note> It is prohibited to set this bit to "10" or "11".
bit27	M5PS: MediaLB 5-pin Select	Sets in order to use MediaLB by 5 pins. Make sure this bit is set to "0" because this LSI does not support 5 pins. Set to "0": 3-pin MediaLB mode Set to "1": 5-pin MediaLB mode. (It is prohibited to set "1".) <Note> Always set this bit to "0".
bit26	MLK: MediaLB Lock	Indicates whether MediaLB is in sync with the Media LB frame (lock). Set to "0": Unlocked state Set to "1": Locked state This bit becomes "1" when FRAMESYNC is received continuously 3 to 5 times at the same frame position. If FRAMESYNC is not received twice continuously from the locked state at the same frame position, this bit becomes "0". <Notes> - When DCCR:MRS is set to "1", this bit is cleared to "0". - When DCCR:MDE is "0", lock is not detected.
bit25	MLE: MediaLB Little endian mode	Sets the type of data to be transmitted/received. Make sure this bit is set to "0" because this LSI does not support the little endian. "0": Big endian mode "1": Little endian mode (It is prohibited to set this mode.) <Note> Always set this bit to "0".
bit24	MHRE: MediaLB Hardware Reset enable	Enables resetting when the reset command is received by the MediaLB frame. Resetting is performed by the resetting request from INIC(Intelligent Network Interface Controller). Resetting is performed when either global (SDCR:MSD=00 _H : all of MediaLB are targeted for resetting) MlbReset (FE _H) or device-specific (SDCR:MSD=DA: MediaLB designated by Device Address is targeted for resetting) MlbReset(FE _H) is received from INIC Set to "0": Resetting is prohibited. Set to "1": Resetting is enabled.
bit23	MRS: MediaLB software Reset	When this bit is set to "1", MediaLB is reset. This bit becomes "0" after the execution of resetting. Set to "0": Normal operation Set to "1": Reset <Notes> - Even when the DCCR:MHRE bit is "1" and resetting operation by the receipt of the reset command is in progress, this bit is reset to "0" after the execution of resetting. - For a read-modify-write instruction, the "0" is read from this bit.
bit22- bit8	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit7- bit0	MDA[8:1]: MediaLB Device Address	Sets Device Address (DA) for the MediaLB device. Device Address is a 16-bit address allocated to identify MediaLB. Device Address is used in the event of the MlbScan command and MlbReset command for the system channel. The received Device Address (DA[15:0]) operates as the command target when DA[15:9] and DA[0] are 0 and DA[8:1] matches this bit.

60.3.2 System Status Configuration Register (SSCR)

System Status Configuration Register (SSCR) is a register to indicate the status of the MediaLB network. SSCR is updated for each MediaLB frame.

■ System Status Configuration Register (SSCR)

Figure 60.3-2 displays the bit configuration of the System Status Configuration Register (SSCR), and Table 60.3-4 displays the function of each bit.

Figure 60.3-2 Bit Configuration of the System Status Configuration Register (SSCR)

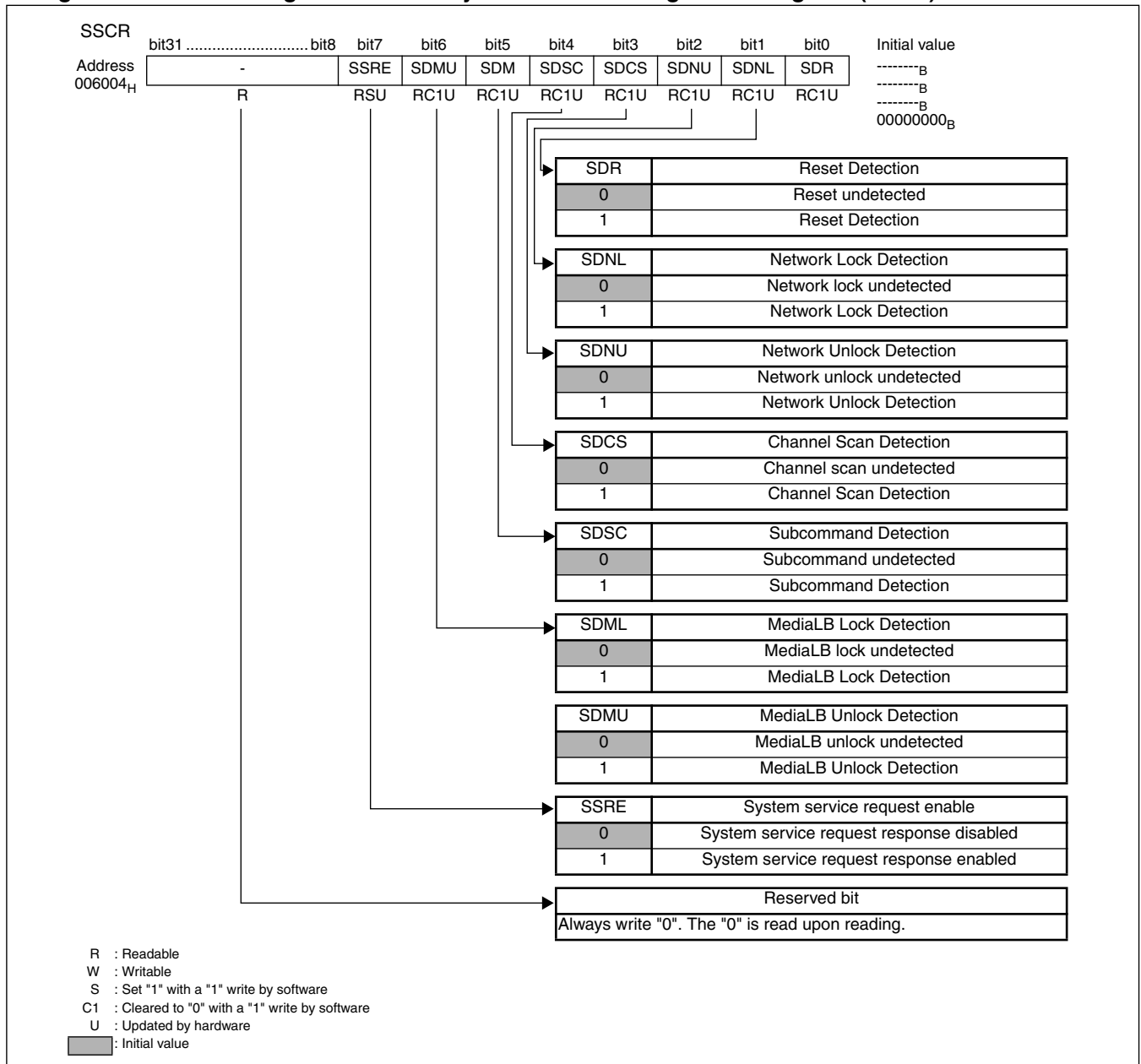


Table 60.3-4 Functional Descriptions on the System Status Configuration Register (SSCR) Bits (1 / 2)

Bit Name		Function
bit31-bit8	rsvd: Reserved	Reserved bit. "0" is read upon reading. Write "0" upon writing.
bit7	SSRE: System service Request Enable	<p>Enables the response to the system service request. When "1" is set for this bit, RxStatus (DeviceServiceRequest(82_H)) is transmitted as a response to the MlbScan (E4_H) system command. When RxStatus is transmitted, this bit is cleared to "0". When "0" is written in this bit, it is invalid.</p> <p><Note> For a read-modify-write instruction, the "0" is read from this bit.</p>
bit6	SDMU: MediaLB unlock Detection	<p>Indicates that unlock was detected from the MediaLB frame. Set to "0": Unlock is not detected. Set to "1": Unlock was detected.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMMU), interrupt occurs as a system interrupt when this bit becomes "1". When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Notes> - During the locked state, this bit is set to "1" when the DCCR:MRS bit is set to "1". - For a read-modify-write instruction, the "0" is read from this bit.</p>
bit5	SDML: MediaLB Lock Detection	<p>Indicates that lock was detected from the MediaLB frame. Set to "0": Lock is not detected. Set to "1": Lock was detected.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMML), interrupt occurs as a system interrupt when this bit becomes "1". When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note> For a read-modify-write instruction, the "0" is read from this bit.</p>
bit4	SDSC: Subcommand Detection	<p>Indicates that the sub command of the system command, MlbSubCmd(E6_H) was received. Set to "0": Sub command is not received. Set to "1": Sub command was received.</p> <p>The details of the command can be confirmed by reading the SDCR register. If not masked by the System Mask Configuration Register (SMCR:SMSC), interrupt occurs as a system interrupt when this bit becomes "1". When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note> For a read-modify-write instruction, the "0" is read from this bit.</p>
bit3	SDCS: Channel Scan Detection	<p>Indicates that the channel scan of the MlbScan(E4_H) system command was received. Set to "0": Channel scan is not received. Set to "1": Channel scan was received.</p> <p>The Device Address of the channel scan can be confirmed by reading the SDCR register. Even if the channel scan is for another device, the Device Address is written in the SDCR register.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMCS), interrupt occurs as a system interrupt when this bit becomes "1". When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note> For a read-modify-write instruction, the "0" is read from this bit.</p>

Table 60.3-4 Functional Descriptions on the System Status Configuration Register (SSCR) Bits (2 / 2)

Bit Name		Function
bit2	SDNU: Network Unlock Detection	<p>Indicates that the network unlock of the system command, MOST_Unlock(E2_H) was received.</p> <p>Set to "0": The network unlock is not received.</p> <p>Set to "1": The network unlock was received.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMNU), interrupt occurs as a system interrupt when this bit becomes "1".</p> <p>When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note></p> <p>For a read-modify-write instruction, the "0" is read from this bit.</p>
bit1	SDNL: Network Lock Detection	<p>Indicates that the network lock of the system command, MOST_Lock(E0_H) was received.</p> <p>Set to "0": The network lock is not received.</p> <p>Set to "1": The network lock was received.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMNL), interrupt occurs as a system interrupt when this bit becomes "1".</p> <p>When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note></p> <p>For a read-modify-write instruction, the "0" is read from this bit.</p>
bit0	SDR: Reset Detection	<p>Indicates that the system command reset, MlbReset(EE_H) was received.</p> <p>Set to "0": Reset is not received.</p> <p>Set to "1": Reset was received.</p> <p>The Device Address of the reset can be confirmed by reading the SDCR register.</p> <p>Even if the reset request is for another device, the Device Address is written in the SDCR register.</p> <p>If not masked by the System Mask Configuration Register (SMCR:SMR), interrupt occurs as a system interrupt when this bit becomes "1".</p> <p>When "1" is written, this bit is cleared to "0". Writing "0" to this bit has no effect.</p> <p><Note></p> <p>For a read-modify-write instruction, the "0" is read from this bit.</p>

<Notes>

- The SSRE, SDSC, SDCS, SDNU, SDNL, and SDR bits in the SSCR register are enabled when MediaLB is in the locked state (DDCR:MLK="1").
- Because the details of the SSCR register status are updated when the next frame is received, check the details of the status before receiving the next frame.

60.3.3 System Data Configuration Register (SDCR)

System Data Configuration Register (SDCR) is a register to receive the system channel data for the MediaLB frame. SDCR is updated for each MediaLB frame.

■ System Data Configuration Register (SDCR)

Figure 60.3-3 displays the bit configuration of the System Data Configuration Register (SDCR), and Table 60.3-5 displays the function of each bit.

Figure 60.3-3 Bit Configuration of the System Data Configuration Register (SDCR)

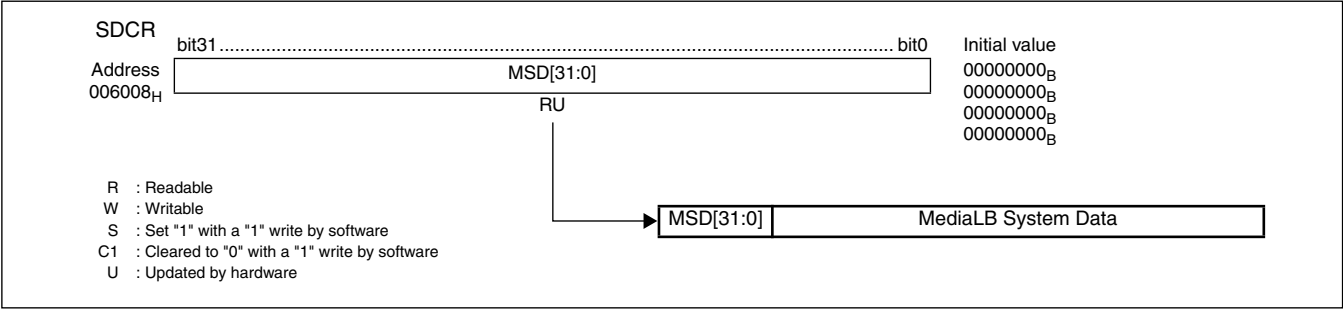


Table 60.3-5 Functional Descriptions on the System Data Configuration Register (SDCR) Bits

Bit Name		Function
bit31-bit0	MSD[31:0]: MediaLB System Data	Four-byte data for the system channel is stored in this register. The received data on the system channel can be read from this register.

<Note>

Read SDCR before the current frame data is lost at the start of the next MediaLB frame.

60.3.4 System Mask Configuration Register (SMCR)

System Mask Configuration Register (SMCR) sets the system interrupt mask.

■ System Mask Configuration Register (SMCR)

Figure 60.3-4 displays the bit configuration of the System Mask Configuration Register (SMCR), and Table 60.3-6 displays the function of each bit.

Figure 60.3-4 Bit Configuration of the System Mask Configuration Register (SMCR)

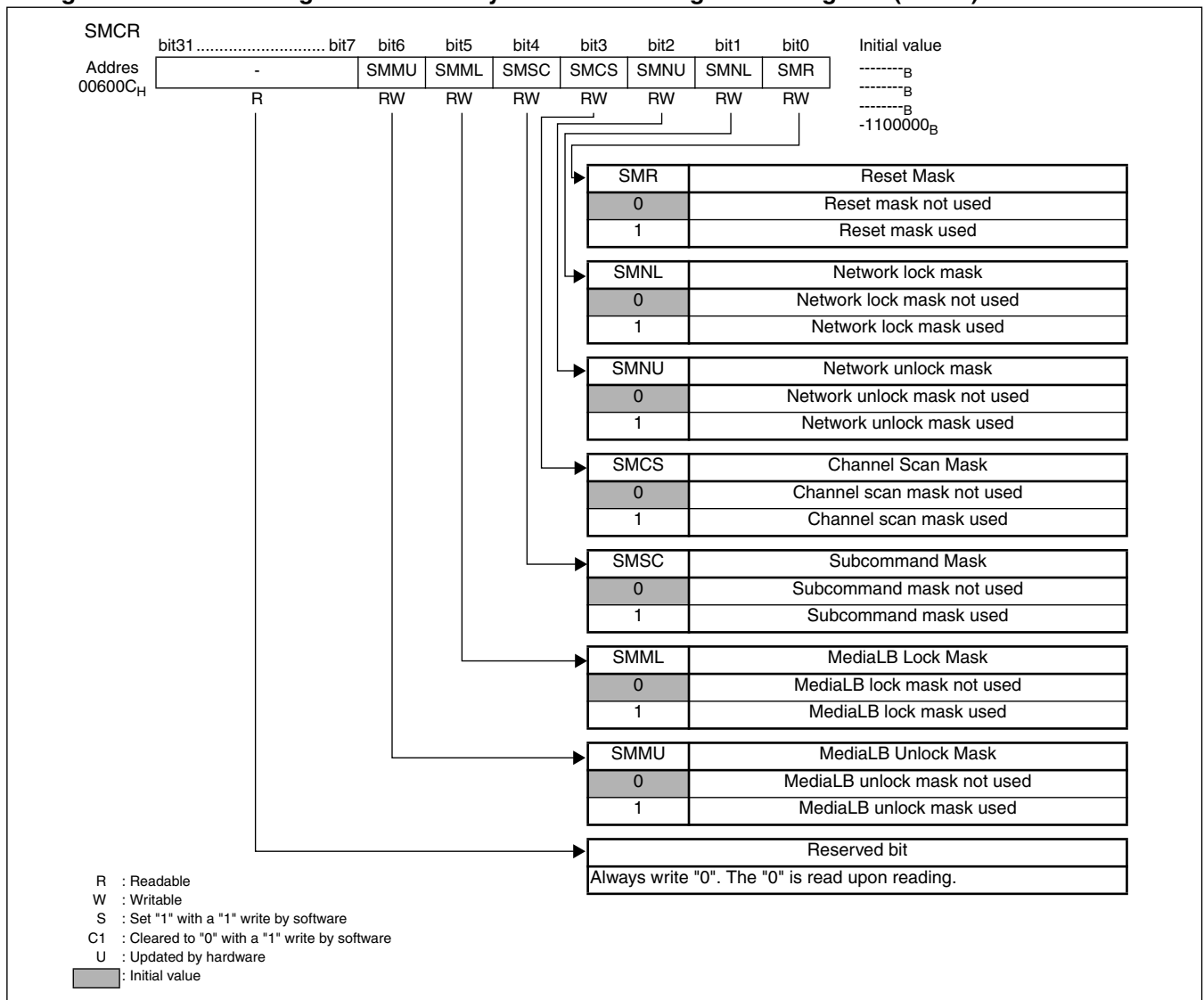


Table 60.3-6 Functional Descriptions on the System Mask Configuration Register (SMCR) Bits

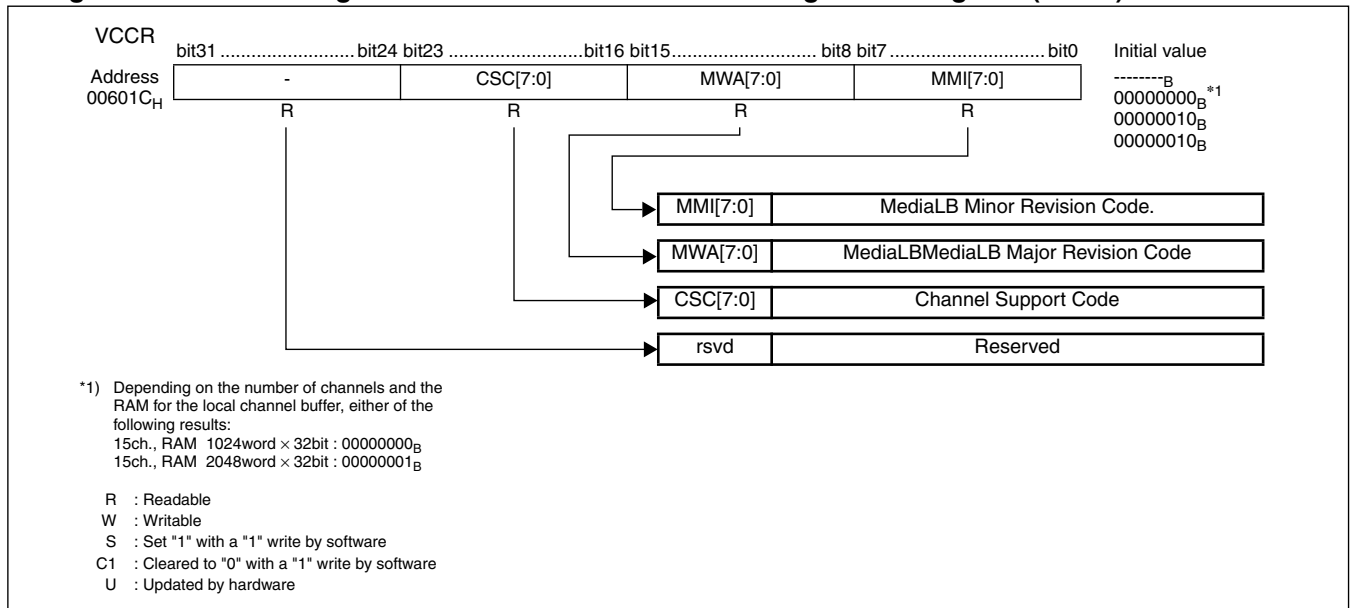
Bit Name		Function
bit31-bit7	rsvd: Reserved	Reserved bit. "0" is read upon reading. Write "0" when writing.
bit6	SMMU: MediaLB Unlock Mask	Sets whether system interrupt should be masked when unlock was detected from the MediaLB frame. The bit targeted for masking is SSCR:SDMU. Set to "0": Unlock interrupt of MediaLB is not masked. Set to "1": Unlock interrupt of MediaLB is masked.
bit5	SMML: MediaLB Lock Mask	Sets whether system interrupt should be masked when lock was detected from the MediaLB frame. The bit targeted for masking is SSCR:SDML. Set to "0": Interrupt in the event of lock detection from MediaB is not masked. Set to "1": Interrupt in the event of lock detection from MediaLB is masked.
bit4	SMSC: Subcommand Mask	Sets whether system interrupt should be masked when the sub command of the system command, MlbSubCmd(E6 _H) was received. The bit targeted for masking is SSCR:SDSC. Set to "0": Interrupt by the receipt of the MlbSubCmd(E6 _H) system command is not masked. Set to "1": Interrupt by the receipt of the MlbSubCmd(E6 _H) system command is masked.
bit3	SMCS: Channel Scan Mask	Sets whether system interrupt should be masked when the system command, channel scan MlbScan(E4 _H) was received. The bit targeted for masking is SSCR:SDCS. Set to "0": Interrupt by the receipt of the MlbScan(E4 _H) system command is not masked. Set to "1": Interrupt by the receipt of the MlbScan(E4 _H) system command is masked.
bit2	SMNU: Network Unlock Mask	Sets whether system interrupt should be masked when the system command, network unlock MOST_Unlock(E2 _H) was received. The bit targeted for masking is SSCR:SDNU. Set to "0": Interrupt by the receipt of the MOST_Unlock(E2 _H) system command is not masked. Set to "1": Interrupt by the receipt of the MOST_Unlock(E2 _H) system command is masked.
bit1	SMNL: Network Lock Mask	Sets system interrupt should be masked when the system command, network lock MOST_Lock(E0 _H) was received. The bit targeted for masking is SSCR:SDNL. Set to "0": Interrupt by the receipt of the MOST_Lock(E0 _H) system command is not masked. Set to "1": Interrupt by the receipt of the MOST_Lock(E0 _H) system command is masked.
bit0	SMR: Reset Mask	Sets whether system interrupt should be masked when the system command, reset MlbReset(FE _H) was received. The bit targeted for masking is SSCR:SDR. Set to "0": Interrupt by the receipt of the MlbReset(FE _H) system command is not masked. Set to "1": Interrupt by the receipt of the MlbReset(FE _H) system command is masked.

60.3.5 Version Control Configuration Register (VCCR)

Version Control Configuration Register (VCCR) is a register to indicate the MediaLB device version.

■ Version Control Configuration Register (VCCR)

Figure 60.3-5 displays the bit configuration of the Version Control Configuration Register (VCCR), and Table 60.3-7 displays the function of each bit.

Figure 60.3-5 Bit Configuration of the Version Control Configuration Register (VCCR)**Table 60.3-7 Functional Descriptions on the Version Control Configuration Register (VCCR) Bits**

Bit Name		Function
bit31-bit24	rsvd: Reserved	Reserved bit. The "0" is read upon reading.
bit23-bit16	CSC[7:0]: Channel Support Code	Indicates the channels to be supported and the RAM size for the local channel buffer. 00 _H : 15 channels; Supports local channel buffer 1024 words × 32 bits. 01 _H : 15 channels; Supports local channel buffer 2048 words × 32 bits.
bit15-bit8	MMA[7:0]: MediaLB Major Revision Code	Indicates the major revision of the MediaLB device. This bit indicates 02 _H .
bit7-bit0	MMI[7:0]: MediaLB Minor Revision Code	Indicates the minor revision of the MediaLB device. This bit indicates 02 _H .

60.3.6 Channel Interrupt Configuration Register (CICR)

Channel Interrupt Configuration Register (CICR) indicates the channel in which channel interrupt is occurring. This bit is cleared to "0" by writing "1" in the interrupt flag for which masking is not set (CSCRn:STS[15:0]).

■ Channel Interrupt Configuration Register (CICR)

Figure 60.3-6 displays the bit configuration of the Channel Interrupt Configuration Register (CICR), and Table 60.3-8 displays the function of each bit.

Figure 60.3-6 Bit Configuration of the Channel Interrupt Configuration Register (CICR)

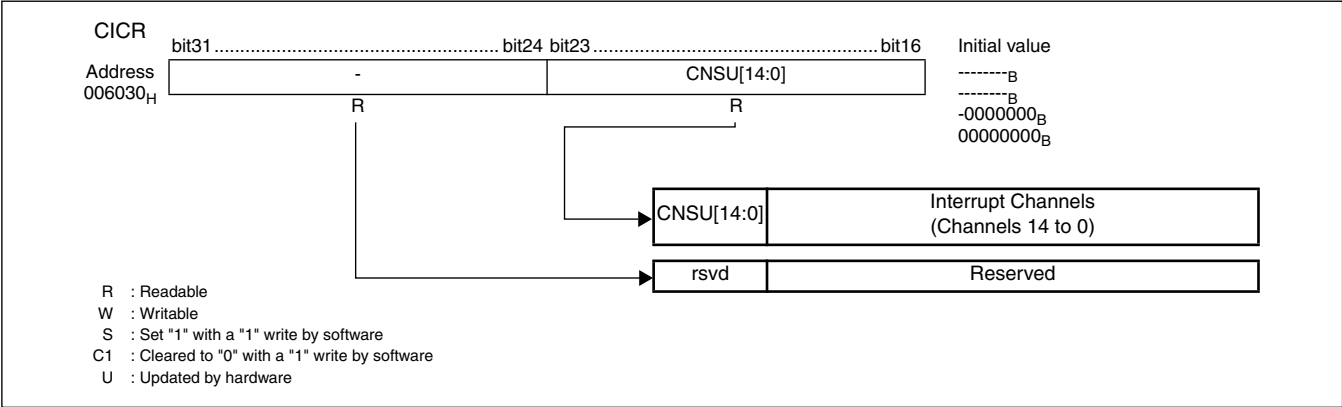


Table 60.3-8 Functional Descriptions on the Channel Interrupt Configuration Register (CICR) Bits

Bit Name		Function
bit31- bit15	rsvd: Reserved	Reserved bit. The "0" is read upon reading.
bit14- bit0	CNSU[14:0]: Interrupt Channels (Channels 14 to 0)	Indicates that channel interrupt occurred. CNSU[0] indicates whether there is an interrupt of channel 0, CNSU[1] indicates whether there is an interrupt of channel 1, and CNSU[14] indicates whether there is an interrupt of channel 14. Set to "0": There is no interrupt. Set to "1": There is an interrupt. Each bit is cleared to "0" by writing "1" in the interrupt flag for which masking is not set (CSCRn:STS[15:0]).

60.3.7 Channel n Entry Configuration Register (CECRn)

Channel n Entry Configuration Register (CECRn) is a register to set the channel enable, channel type, channel direction, channel control enable, channel mode, channel interrupt mask, and channel address of the logical channel.

■ Channel n Entry Configuration Register (CECRn)

Figure 60.3-7 displays the bit configuration of the Channel n Entry Configuration Register (CECRn), and Table 60.3-9 displays the function of each bit.

Figure 60.3-7 Bit Configuration of the Channel n Entry Configuration Register (CECRn)

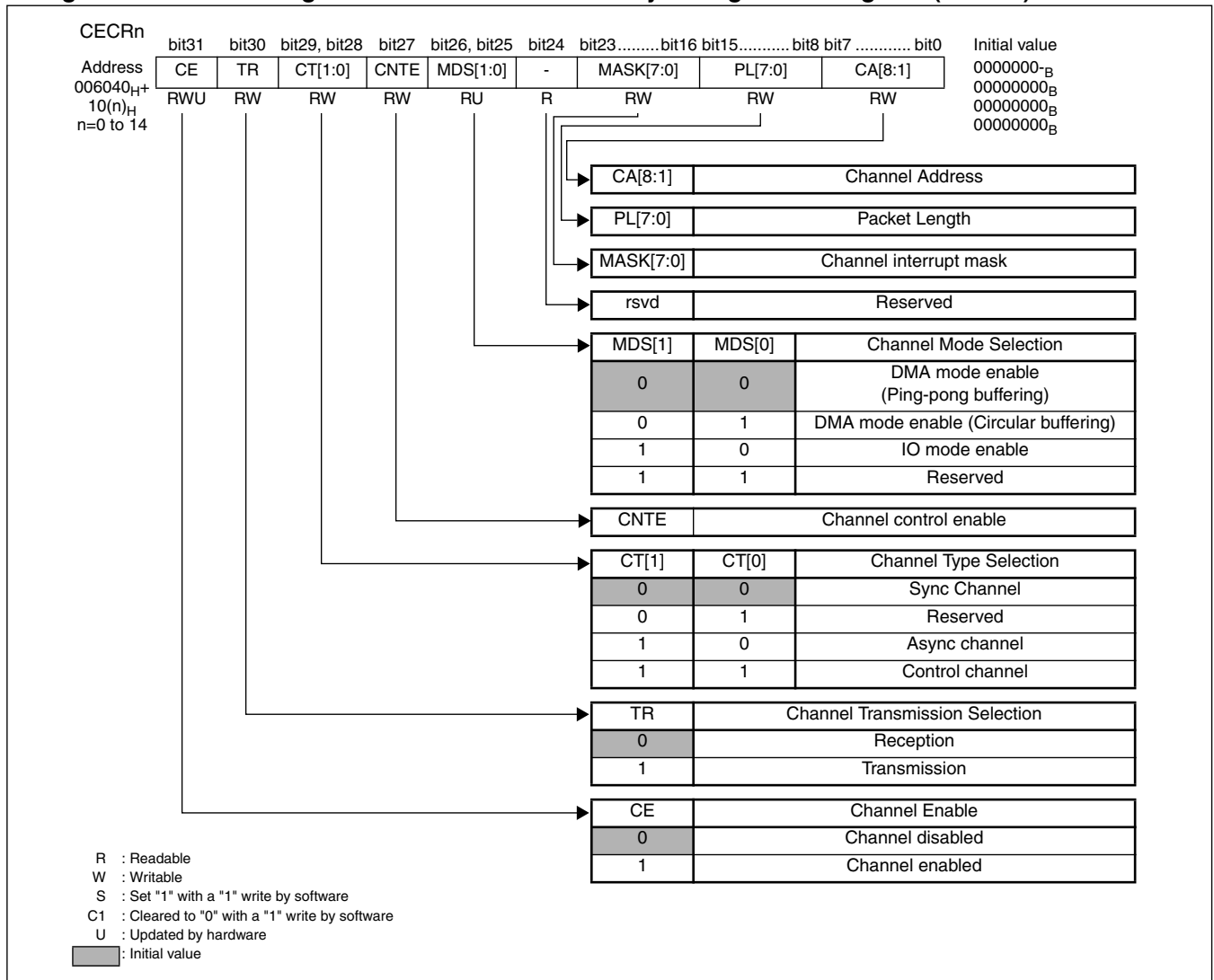


Table 60.3-9 Functional Descriptions on the Channel n Entry Configuration Register (CECRn) Bits (1 / 4)

Bit Name		Function
bit31	CE: Channel Enable	Enables channels. Set to "0": Channels are prohibited. Set to "1": Channels are enabled. This bit is reset to "0" when "1" is set for CECRn:PL[7] and a frame sync lost error is detected in the case of the sync channel (CECR:CT="00").
bit30	TR: Channel Transmission Selection	Sets whether the channel is for transmission or reception. Set to "0": Reception Set to "1": Transmission
bit29, bit28	CT[1:0]: Channel Type Selection	Sets the channel type selection. Set to "00": is set: Sync channel Set to "01": Reserved Set to "10": Async channel Set to "11": is set: Control channel

Table 60.3-9 Functional Descriptions on the Channel n Entry Configuration Register (CECRn) Bits (2 / 4)

Bit Name			Function
bit27	CNTE: Channel Control Enable	Async Channel/ Control Channel	Sets whether the reception packet counter should be enabled. Set to "0": Reception packet counter disable Set to "1": Reception packet counter enable <Note> Valid only in IO mode.
		Sync Channel	Sets whether the frame sync of the streaming channel should be enabled. Set to "0": Frame sync disable Set to "1": Frame sync enable
bit26, bit25	MDS[1:0]: Channel Mode Selection		Sets the channel mode. Set to "00": Ping-pong buffering (DMA mode) Set to "01": Circular buffering (DMA mode) Set to "10": IO mode It is prohibited to set "11". <Notes> - Set either the DMA mode or IO mode for the channel to be used. It is prohibited to set the IO mode and DMA mode in a mixed manner for respective channels. - Setting this bit to "11" is prohibited.
bit24	rsvd: Reserved		Reserved bit. The "0" is read upon reading. Write "0" when writing.

Table 60.3-9 Functional Descriptions on the Channel n Entry Configuration Register (CECRn) Bits (3 / 4)

Bit Name			Function
bit23-bit16	Channel interrupt mask	MASK[7]: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
		MASK[6]: Frame Sync Lost Mask	Sets whether channel interrupt by the frame sync lost should be masked. The status bit targeted for masking is SCSRn:STS[6]. Set to "0": No interrupt mask Set to "1": With interrupt mask
		MASK[5]: Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
		MASK[4]: Buffer Error Mask	Sets whether the channel interrupt by a buffer error should be masked. The status bit targeted for masking is SCSRn:STS[4]. Set to "0": No interrupt mask Set to "1": With interrupt mask
		MASK[3]: Buffer Start Mask/ Transmission Service Request Mask	This bit changes its meaning as follows depending on DMA mode/IO mode setting. Set to the DMA mode: Buffer start mask Set to the IO mode: Transmission service request mask This bit sets whether or not to mask channel interrupts. The status bit targeted for masking is SCSRn:STS[3]. Set to "0": No interrupt mask Set to "1": With interrupt mask
		MASK[2]: Buffer End Mask/ Reception Service Request Mask	This bit changes its meaning as follows depending on DMA mode/IO mode setting. Set to the DMA mode: Buffer end mask Set to the IO mode: Reception service request mask or reception packet abort mask This bit sets whether or not to mask channel interrupts. The status bit targeted for masking is SCSRn:STS[2]. Set to "0": No interrupt mask Set to "1": With interrupt mask
		MASK[1]: Break Detection Mask	Sets whether the channel interrupt by the detection of break should be masked. The status bit targeted for masking is SCSRn:STS[1]. Set to "0": No interrupt mask Set to "1": With interrupt mask
		MASK[0]: Protocol Error Mask	Sets whether the channel interrupt by a protocol error should be masked. The status bit targeted for masking is SCSRn:STS[0]. Set to "0": No interrupt mask Set to "1": With interrupt mask

Table 60.3-9 Functional Descriptions on the Channel n Entry Configuration Register (CECRn) Bits (4 / 4)

Bit Name			Function
bit15-bit8	PL[7:0]: Packet Length	Async Channel/ Control Channel	<p>Sets the packet count threshold.</p> <p>When the reception packets are set in this bit and the number of the packets has reached this setting, a reception service request is generated. Also, when the local channel buffer has become full, a reception service request is generated.</p> <p><Notes></p> <ul style="list-style-type: none"> - When using this bit, make the setting as LCBCRn:TH[9:0]=000_H. - Set "000" for PL[7:5]. - Valid only in the IO mode.
		Sync Channel	<p>PL[7] sets frame synchronous disable.</p> <p>Set to "0": Frame sync channel disable is prohibited.</p> <p>Set to "1": Frame sync channel disable is enabled.</p> <p>If frame synchronous channel disable is enabled, the channel enable bit (CECRn:CE) is reset to "0" when frame sync lost occurs.</p> <p>PL[6:0] sets the frame synchronous physical channels.</p> <p>In other words, the number of physical channels which matches the channel address (CECRn:CA[8:1]) is set among the synchronous channels contained in one frame.</p>
bit7-bit0	CA[8:1]: Channel Address		<p>Sets the channel address of the logical channel.</p> <p>This bit is compared with the channel address of the received physical channel, and data is transmitted/received when they match each other. The received channel address (CA[15:0] is transmitted/received when CA[15:9] and CA[0] are 0 and CA[8:1] match this bit.</p>

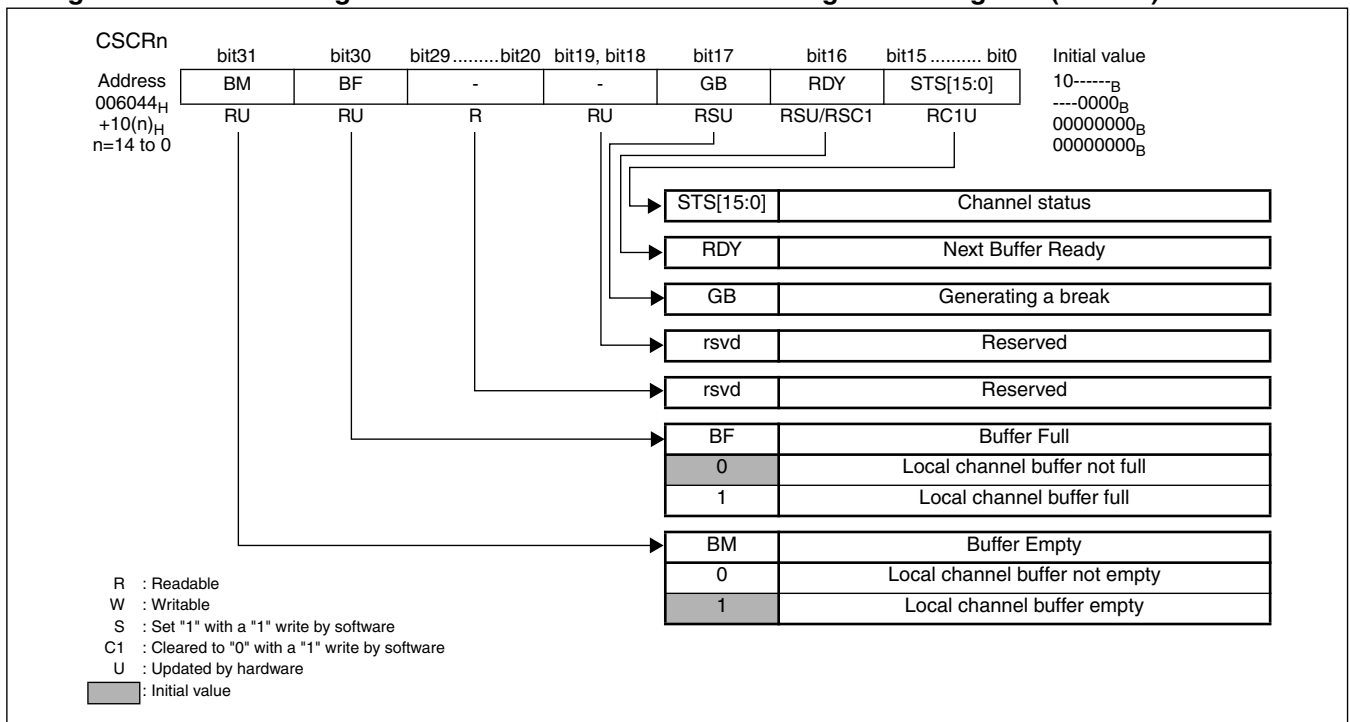
60.3.8 Channel n Status Configuration Register (CSCRn)

Channel n Status Configuration Register (CSCRn) reflects the statuses of the current buffer and previous buffer.

■ Channel n Status Configuration Register (CSCRn)

Figure 60.3-8 displays the bit configuration of the Channel n Status Configuration Register (CSCRn), and Table 60.3-10 displays the function of each bit.

Figure 60.3-8 Bit Configuration of the Channel n Status Configuration Register (CSCRn)



**Table 60.3-10 Functional Descriptions on the Channel n Status Configuration Register (CSCRn) Bits
(1 / 4)**

Bit Name		Function
bit31	BM: Buffer Empty	This bit becomes "1" when the local channel buffer of the channel is empty.
bit30	BF: Buffer Full	This bit becomes "1" when the local channel buffer of the channel is full.
bit29- bit20	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit19, bit18	rsvd: Reserved	Reserved bit. Undefined upon reading. Upon writing, operations are not influenced even if undefined data is written.
bit17	GB: Break Generation	Async Channel/ Control Channel
		Sync Channel

MB91460 Series**Table 60.3-10 Functional Descriptions on the Channel n Status Configuration Register (CSCRn) Bits
(2 / 4)**

Bit Name		Function
bit16	RDY: Next Buffer Ready	<p>For the IO mode, this bit becomes a reserved bit. The "0" is read upon reading. Write "0" when writing.</p> <p>For the DMA mode, this bit becomes a DMA mode start bit. If "1" is set for this bit, data is transferred from the FIFO buffer to the local channel buffer in the case of transmission. For reception, data is transferred from the local channel buffer to the FIFO buffer when received data is stored in the local channel buffer.</p> <p>For ping-pong buffering, this bit is cleared to "0" when data in the next buffer configuration register is copied to the current buffer configuration register.</p> <p>For circular buffering, this bit is not cleared to "0" by the hardware if "1" is set for this bit. To stop circular buffering, write "0" in this bit. Circular buffering is stopped when CCBCRn:BCA reaches CCBCRn:BFA.</p> <p><Note> During ping-pong buffering, packets can be transmitted/received continuously if "1" is written in this bit after this bit is cleared to "0" by the hardware and the next packet is set. However, such a usage is prohibited unless the capacity of the FIFO buffer enough for the packet amount is secured.</p>
bit15- bit10	Channel status	STS[15:12]: Reserved
		STS[11]: Reserved/ Previous Buffer Start
		STS[10]: /Reserved/ Previous Buffer End

Table 60.3-10 Functional Descriptions on the Channel n Status Configuration Register (CSCRn) Bits
(3 / 4)

Bit Name			Function
bit9-bit7	Channel status	STS[9]: Reception Packet Start/ Previous Buffer Break	<p>During the IO mode, this bit becomes a reception packet start bit. This bit indicates that the reception channel detected the packet start command, ControlStart(30_H) or AsyncStart(20_H)</p> <p>Set to "0": Reception packet start command is not detected. Set to "1": Reception packet start command is detected.</p> <p>This bit is used to detect that the reception packet was aborted and the next packet reception was started. When this bit becomes "1", processing of valid data can be started.</p> <p>During the DMA mode, this bit becomes a previous buffer break bit. This bit is set to "1" in the following conditions.</p> <ul style="list-style-type: none"> - The current buffer break is set to "1" at the start of the current buffer. - In the transmission setting, the local channel buffer contains data transferred in the previous DMA mode. ReceiverBreak is received during the next DMA mode processing. <p>When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect.</p> <p><Notes></p> <ul style="list-style-type: none"> - During the IO mode, this bit enables only the transmission of the asynchronous and control channels. - During the IO mode, channel interrupt does not occur even if this bit is set to "1". This bit must be checked periodically following the reception packet abort (STS[8]). - For a read-modify-write instruction, the "0" is read from this bit.
		STS[8]: Reception Packet Abort/ Previous Buffer Protocol Error	<p>During the IO mode, this bit becomes a reception packet abort bit. In the reception setting, "1" is set for this bit when there is a ReceiverBreak(70_H) response or when ControlBreak(36_H) or AsyncBreak(26_H) is received.</p> <p>During the DMA mode, this bit becomes a previous buffer protocol error bit. When the current DMA mode processing ends while the current buffer protocol error is set to "1", when a invalid command is received in the reception setting, or when ReceiverBreak(70_H) is received halfway through the packet in the reception or transmission setting for the ControlStart(30_H) or AsyncStart(20_H) command, the RDY bit is cleared to "0" if the current DMA mode processing ends while the RDY bit is "1", then, this bit is set to "1" and the next DMA mode processing is started.</p> <p>When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect.</p> <p><Note></p> <p>For a read-modify-write instruction, the "0" is read from this bit.</p>
		STS[7]: Reserved	Reserved bit. The "0" is read upon reading. Write "1" when writing.
bit6	Channel status	STS[6]: Frame Sync Lost	<p>Frame sync lost bit. If synch to the MediaLB frame is lost, "1" is set for this bit. When CECRn:CNTS is set to "1" and the physical channels set in CECRn:PL[6:0] do not match the physical channels of which channel address is matched, such a state is regarded as sync lost.</p> <p>When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect.</p> <p><Notes></p> <ul style="list-style-type: none"> - For a read-modify-write instruction, the "0" is read from this bit. - This bit is valid only in the sync channel.

MB91460 Series

Table 60.3-10 Functional Descriptions on the Channel n Status Configuration Register (CSCnRn) Bits
(4 / 4)

Bit Name			Function
bit5-bit0	Channel status	STS[5]: Reserved/ Host Bus Error	Reserved bit in IO mode. The "0" is read upon reading. Write "1" when writing. During the DMA mode, this bit becomes a host bus error bit. When the MediaLB direction and the FIFO buffer direction are incorrect or writing from MediaLB occurs while the FIFO buffer channel is full, this bit is set to "1" if BCA of the current buffer configuration register (CCBCRn) is out of the address area set in the FIFO Buffer Channel n Address Range Register (BUFARn). When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.
		STS[4]: Buffer Error	Buffer error bit. This bit is set to "1" if a local channel buffer underrun error in the transmission setting or local channel buffer overrun error in the reception setting occurs. When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.
		STS[3]: Transmission Service Request/ Current Buffer Start	During the IO mode, this bit becomes a transmission service request bit. This bit is set to "1" if a valid transmission data in the local channel buffer reaches below LCBCRn:TH[8:0] in the transmission setting. During the DMA mode, this bit become a current buffer start bit. When the CNBCRn setting value is transferred to CCBCRn, the RDY bit is set to "0" and this bit to "1", and the DMA mode processing is started. In DMA mode, this bit is cleared to "0" when "1" is written. Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.
		STS[2]: Reception Service Request/ Current Buffer End	During the IO mode, this bit becomes a reception service request bit. If the quadlets in which the local channel buffer is empty reaches below LCBCRn:TH[8:0] in the reception setting, "1" is set for this bit. During the DMA mode, this bit becomes a current buffer end bit. If the transmission/reception of the final quadlet in the final packet was successful, "1" is set for this bit and the DMA mode processing ends. In DMA mode, this bit is cleared to "0" when "1" is written. Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.
		STS[1]: Current Buffer Break	Current buffer break bit. When ReceiverBreak(70 _H) is received in the transmission setting or when ControlBreak(36 _H) or AsyncBreak(26 _H) is received in the reception setting, "1" is set for this bit. When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.
		STS[0]: Current Buffer Protocol Error	Current buffer protocol error. When an invalid command is received in the reception setting, when ControlStart(30 _H) or AsyncStart(20 _H) command is received halfway through the packet, or when ReceiverBreak(70 _H) is received in the transmission setting, "1" is set for this bit. When "1" is written to this bit, it is cleared to "0". Writing "0" has no effect. <Note> For a read-modify-write instruction, the "0" is read from this bit.

60.3.9 Channel n Current Buffer Configuration Register (CCBCRn)

If the DMA mode is set for the Channel n Current Buffer Configuration Register (CCBCRn), this register indicates the current address. When the IO mode is set, the register becomes a reception data buffer.

■ Channel n Current Buffer Configuration Register (CCBCRn)

The function of this register varies in the operations of the MediaLB in IO mode and in DMA mode.

● Channel n Current Buffer Configuration Register during the IO Mode

Figure 60.3-9 displays the bit configuration of the Channel n Current Buffer Configuration Register (CCBCRn), Table 60.3-11 displays the function of each bit.

Figure 60.3-9 Bit Configuration of the Channel Current Buffer Configuration Register (CCBCRn) During the IO Mode

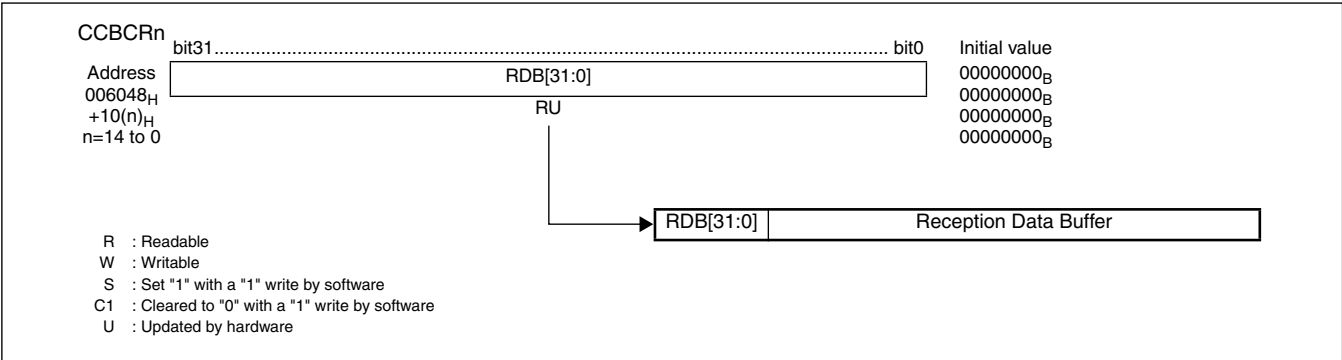


Table 60.3-11 Functional Descriptions on the Channel n Current Buffer Configuration Register (CCBCRn) Bits during the IO Mode

Bit Name		Function
bit31-bit0	RDB[31:0]: Reception Data Buffer	When the logical channel is under reception, the reception data is read out by reading this register. <Note> To read his register, read it out by 32-bit access.

● Channel n Current Buffer Configuration Register during the DMA Mode

Figure 60.3-10 displays the bit configuration of the Channel n Current Buffer Configuration Register (CCBCRn), and Table 60.3-12 displays the function of each bit.

Figure 60.3-10 Bit Configuration of the Channel n Current Buffer Configuration Register (CCBCRn)

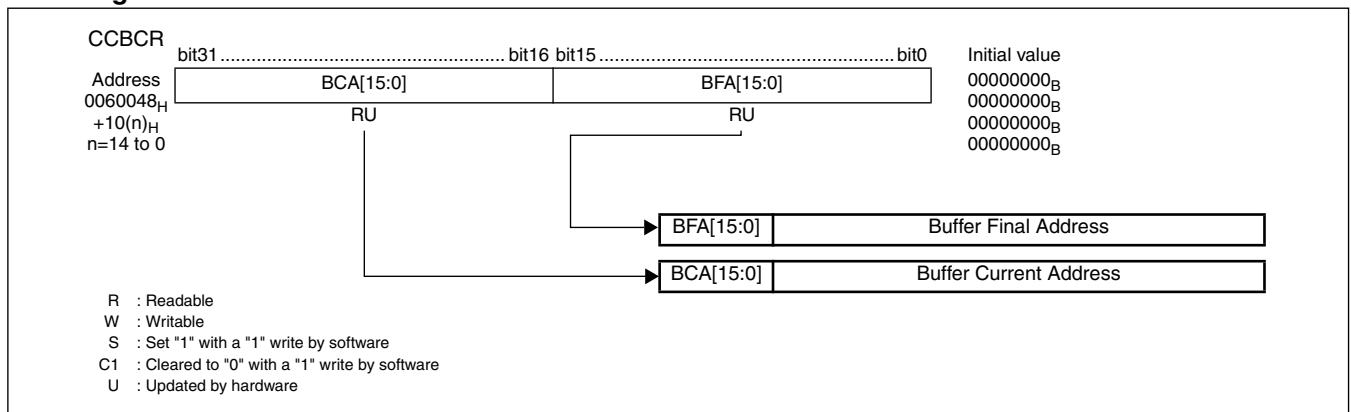
during the DMA Mode

Table 60.3-12 Functional Descriptions on the Channel Current Buffer Configuration Register (CCBCRn) Bits during the DMA Mode

Bit Name		Function
bit31-bit16	BCA[15:0]: Buffer Current Address	<p>Indicates the current address accessed to the FIFO buffer. When "1" is set for the CSCRn:RDY bit, the DMA mode is started, the setting value for CNBCRn:BSA is copied to this bit, and 4 is added every time the logical channel transmits/receives data.</p> <p><Notes></p> <ul style="list-style-type: none"> - Fixed to BCA[1:0]= "00" - During the transmission setting for ping-pong buffering in the control channel or async channel, the remaining packet is discarded when the channel address is received next if the AsyncBreak(26_H) or ControlBreak(36_B) command is output halfway through the packet. At that time, if the final address of the current buffer configuration register is reached, the DMA mode is ended. If the final address is not yet reached, the next packet transmission is started. - During the reception setting for ping-pong buffering in the control channel or async channel, the DMA mode is not stopped even if ReceiveBreak(70_H) is output as RxStatus. If AsyncStart (20_H) or ControlStart (30_H) is received again, data is stored in the FIFO buffer from the start address of the current buffer. Make sure to read the data that has been received by then from the FIFO buffer.
bit15-bit0	BFA[15:0]: Buffer Final Address	<p>Indicates the final address accessed to the FIFO buffer. When "1" is set for the CSCRn:RDY bit, the DMA mode is started and the setting value for CNBCRn:BEA is copied to this bit.</p> <p><Note></p> <p>Fixed to BFA[1:0]= "00"</p>

60.3.10 Channel n Next Buffer Configuration Register (CNBCRn)

If the DMA mode is set for the channel n next buffer configuration register (CNBCRn), this register is used to set the start address and end address of the next address in logical channel n. If the IO mode is set, the CNBCRn register becomes a transmission data buffer.

■ Channel n Next Buffer Configuration Register (CNBCRn)

The function of this register varies in the operations of the MediaLB in IO mode and in DMA mode.

● Channel n Next Buffer Configuration Register During IO Mode

Figure 60.3-11 shows the bit configuration of the channel n next buffer configuration register (CNBCRn) and Table 60.3-13 shows the functions of the bits.

Figure 60.3-11 Bit Configuration of the Channel n Next Buffer Configuration Register (CNBCRn) during the IO Mode

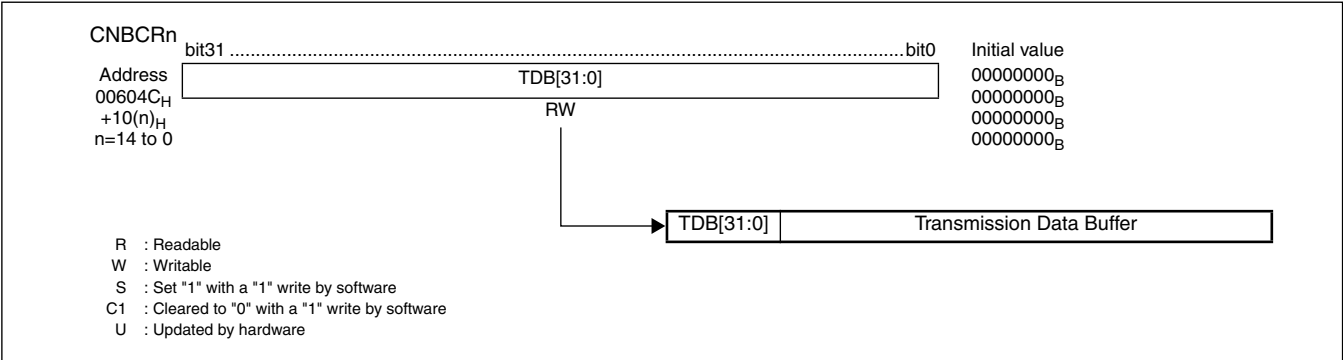


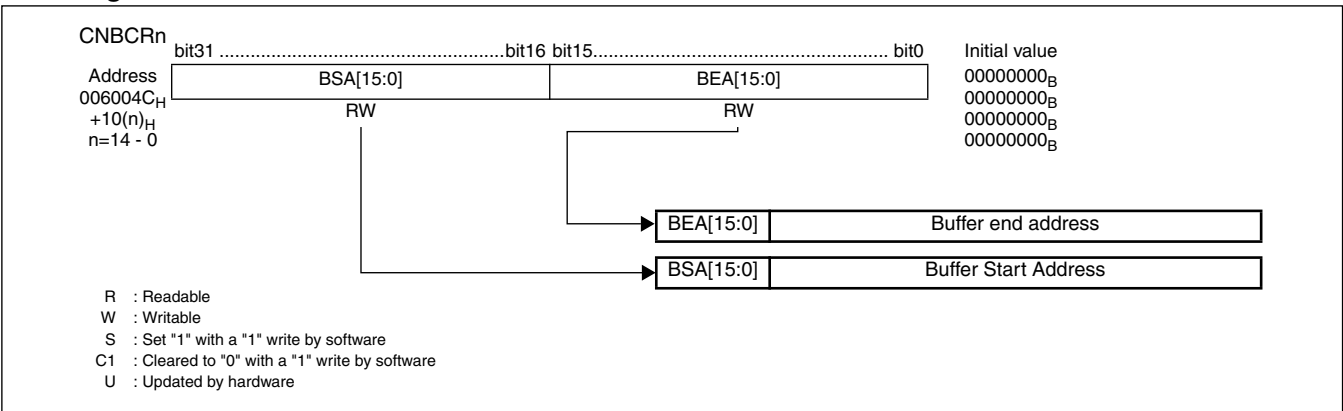
Table 60.3-13 Functional Descriptions on the Channel n Next Buffer Configuration Register (CNBCRn) Bits during the IO Mode

Bit Name		Function
bit31-bit0	TDB[31:0]: Transmission Data Buffer	<p>Transmission data buffer register</p> <p>If transmission is set for the logical channel, written data is output onto the MediaLB frame when the received channel address matches the channel address set for the logical channel.</p> <p><Notes></p> <ul style="list-style-type: none">- To transmit the same data twice, set "1" for the I2SCCR:DBL bit (twice transfer mode) and then write the transmission data once.- The twice transfer mode supports only logical channels 8 and 9.- When writing in this register, use 32 bits if the stereo mode (MSTD:MSTDn="0") is set. If the monaural mode (MSTD:MSTDn="1") is set, write by 16 bits. If address 1 is "L" "0" is written in TDB[31:16], and 16 bit transmission data is written in TDB[15:0]. If address 1 is "H", 16 bit transmission data is written in TDB[31:16], and "0" is written in TDB[15:0].

● Channel Next Buffer Configuration Register during the DMA Mode

Figure 60.3-12 shows the bit configuration of the channel n next buffer configuration register (CNBCRn) and Table 60.3-14 shows the functions of the bits.

Figure 60.3-12 Bit Configuration of the Channel n Next Buffer Configuration Register (CNBCRn) during the DMA Mode



MB91460 Series**Table 60.3-14 Functional Descriptions on the Channel n Next Buffer Configuration Register (CNBCRn) Bits during the DMA Mode**

Bit Name		Function
bit31-bit16	BSA[15:0]: Buffer Start Address	<p>Sets the start address corresponding to the FIFO buffer. After this register is set while CSCRn:RDY bit is "0", the value of this register is transferred to CCBCRn to start DMA mode when "1" is written to CSCRn:RDY bit, if the DMA mode operation is being stopped.</p> <p><Notes></p> <ul style="list-style-type: none"> - BSA[1:0] (bits 17 and 16) is a reserved bit. - Set this register so that it is within the area set in BUFARn. BUFARn indicates the address area of channel n in the FIFO buffer. <p>Example) To make access between channel 0 in the FIFO buffer and the MediaLB device BUFAR0:ST[15:2] £ CNBCR2:BSA[15:2], BUFAR0:EA[15:2] CNBCR2:BEA[15:2]</p> <p>Make the setting which satisfies the following conditions: However, data may be received beyond the buffer end address (BEA). Make some allowance over the buffer end address when setting BUFAR0:EA. If data exceeds the BUFAR0:EA setting, data in the other channel of the FIFO buffer is destroyed or incorrect data is transmitted.</p>
bit15-bit0	BEA[15:0]: Buffer End Address	<p>Sets the end address corresponding to the FIFO buffer. After this register is set while CSCRn:RDY bit is "0", the value of this register is transferred to CCBCRn to start DMA mode when "1" is written to CSCRn:RDY bit, if the DMA mode operation is being stopped.</p> <p><Notes></p> <ul style="list-style-type: none"> - Fixed to BEA[1:0]='00' - For other precautions, refer to the BSA bit.

<Notes>

- If the monaural mode (MSTD:MSTDn="1") is used, write "0" in all MSTD registers before writing in this register.
- During the transmission setting, make the setting for the Channel n Next Buffer Configuration Register so as to match the amount of transmission data.
- During the reception setting, data may be received beyond the buffer end address (BEA). Make some allowance over the buffer end address when setting BUFARn:EA.

60.3.11 Local Channel n Buffer Configuration Register (LCBCRn)

Local Channel n Buffer Configuration Register (LCBCRn) is a register to make the setting for the allocation of the local channel buffer area on the RAM for the local channel buffer. Set this register when all logical channels are disabled.

■ Local Channel n Buffer Configuration Register (LCBCRn)

Figure 60.3-13 displays the bit configuration of the Local Channel n Buffer Configuration Register (LCBCRn), and Table 60.3-15 displays the function of each bit.

Figure 60.3-13 Bit Configuration of the Local Channel n Buffer Configuration Register (LCBCRn)

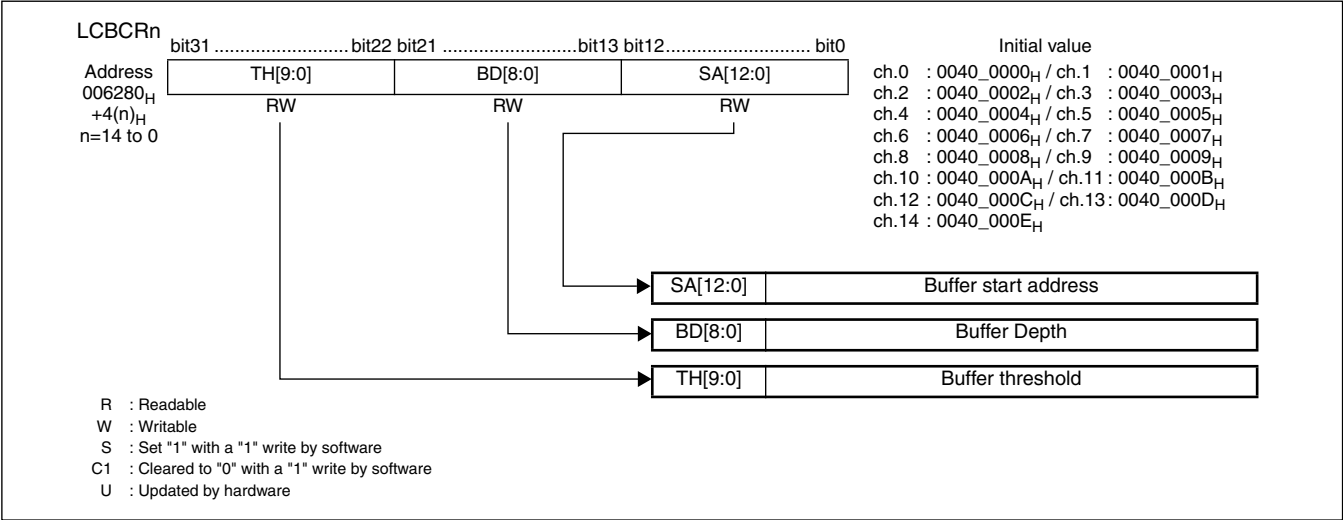
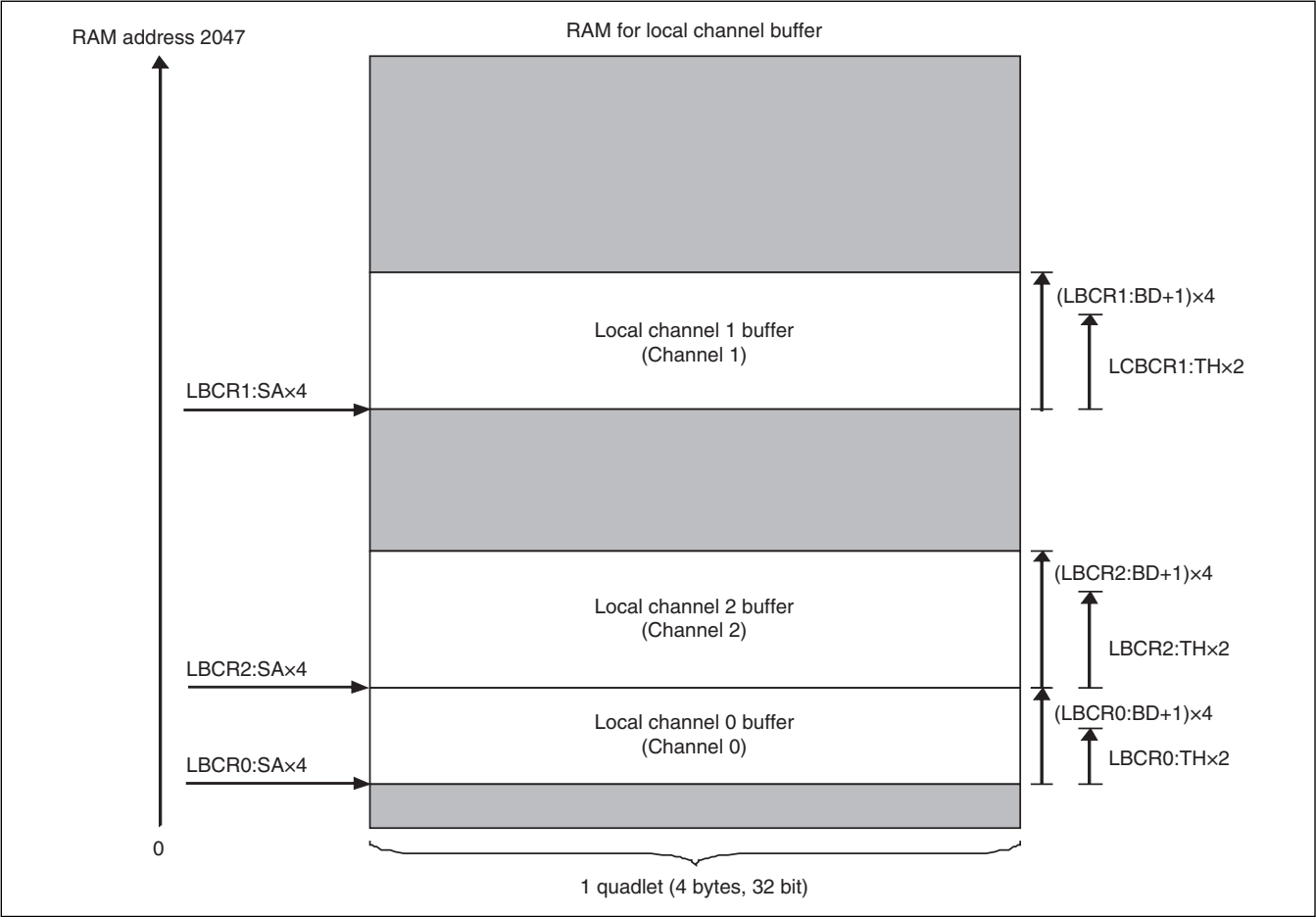


Table 60.3-15 Functional Descriptions on the Local Channel n Buffer Configuration Register (LCBCRn) Bits

Bit Name		Function
bit31-bit22	TH[9:0]: Buffer Threshold (Quadlet/2)	<p>Sets the threshold of the local channel buffer. The buffer threshold is set in units of 2 quadlets. This bit is used to determine whether a transmission/reception service request should be issued. A reception service request is generated when 000_H: TH[9:0] is 000_H and the buffer is full, and a transmission service request is generated when the buffer is empty. 001_H: Threshold=2 quadlets 002_H: Threshold=4 quadlets ... 1FF_H: Threshold=1022 quadlets 200_H: Threshold=1024 quadlets 201_H: Threshold=1026 quadlets ... 3FF_H: Threshold=2046 quadlets <Notes> <ul style="list-style-type: none"> - Valid only in IO mode. - During the reception setting, set "0" for this bit to use the CECRn:PL bit. - For the 1024 word RAM, the TH[9] setting is invalid. </p>
bit21-bit13	BD[8:0]: Buffer Depth ((Quadlet /4)-1)	<p>Sets the depth of the local channel buffer. The depth is set in units of 4 quadlets. 000_H: Depth= 4 quadlets 001_H: Depth= 8 quadlets 002_H: Depth= 12 quadlets ... 0FF_H: Depth= 1024 quadlets 100_H: Depth= 1028 quadlets ... 1FF_H: Depth= 2048 quadlets <Note> For the 1024 word RAM, the BD[8] setting is invalid.</p>
bit12-bit0	SA[12:0]: Buffer Start Address (quadlet/ 4)	<p>Sets the start address of the local channel buffer. The actual RAM address becomes SA ¥ 4 quadlets. 000_H: Start address= 0 quadlet 001_H: Start address= 4 quadlets 002_H: Start address= 8 quadlets ... 0FF_H: Start address= 1020 quadlets 100_H: Start address= 1024 quadlets ... 1FF_H: Start address= 2044 quadlets <Note> For the 1024 word RAM, the SA[12:8] setting is invalid. For the 2048 word RAM, the SA[12:9] setting is invalid.</p>

Figure 60.3-14 displays the relationship between the RAM for the local channel buffer and the Local Channel Buffer Configuration register (LCBCRn).

Figure 60.3-14 LCBC Example of the Rn Setting



60.3.12 FIFO Buffer Channel n Address Range Register (BUFARn n=0 to 14)

Channel n Address Range Register (BUFARn) of the FIFO buffer is a register to set the address range of each channel in the FIFO buffer. It is necessary to set this register when using MediaLB in the DMA mode. If within the setting range of this register as a result of comparison with the address on the HBI bus, access to the corresponding FIFO buffer channel is enabled. The start address is set for ST[15:2] and the end address is set for EA[15:2].

■ FIFO Buffer Channel n Address Range Register (BUFARn)

Figure 60.3-15 displays the bit configuration of the FIFO Buffer Channel n Address Range Register (BUFARn), and Table 60.3-16 displays the function of each bit.

Figure 60.3-15 Bit Configuration of the FIFO Buffer Channel n Address Range Register (BUFARn)

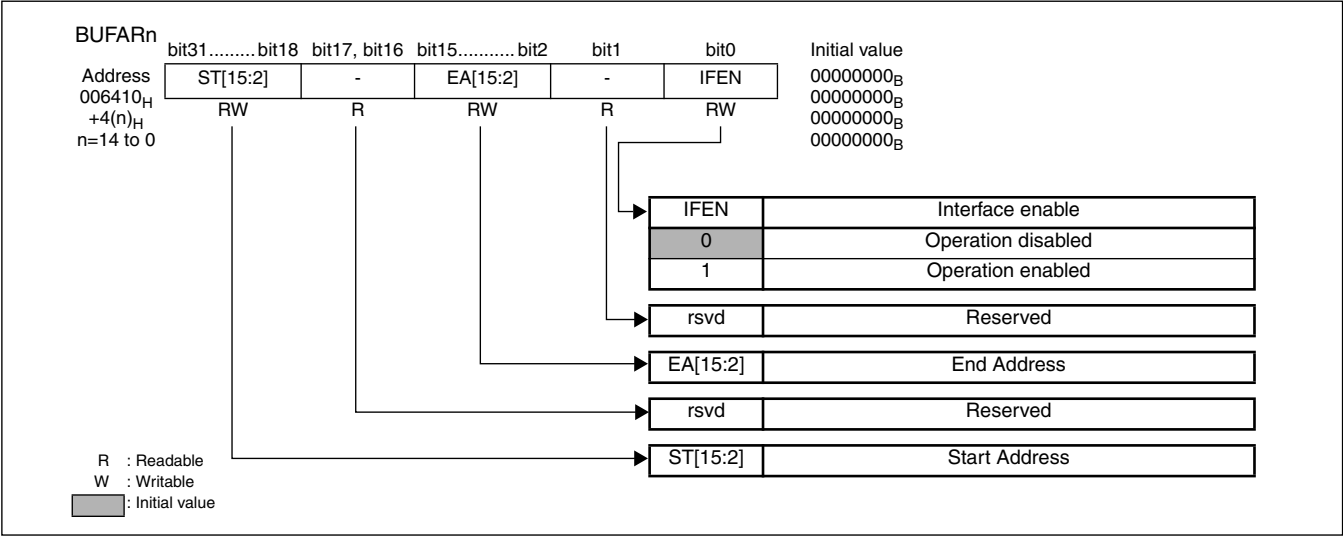


Table 60.3-16 Functional Descriptions on the FIFO Buffer Channel n Address Range Register (BUFARn) Bits

Bit Name		Function
bit31-bit18	ST[15:2]: Start Address	Sets the channel start address of the FIFO buffer. <Note> ST[1:0] is fixed to "00".
bit17, bit16	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit15-bit2	EA[15:2]: End Address	Sets the channel end address of the FIFO buffer. <Note> EA[1:0] is fixed to "00".
bit1	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit0	IFEN: Interface Enable	<p>Enables the area determination operation of the interface. Set to "1": Operation is enabled. Set to "0": Operation is prohibited.</p> <p>When "1" is set for this bit and an access to the FIFO buffer is made by the DMA mode, CCBCRn:BCA[15:0] is output to the interface to determine whether the address is within the area set in this register (ST[15:2] BCA[15:2] EA[15:2]). If it is determined that the address is within the area, the FIFO buffer channel within the area is accessed if the FIFO buffer channel is out of the target area, the following operation is performed.</p> <p>Reception: received data is discarded and "1" is set for the CSCRN:STS[5] bit. Transmission: 0 is output as transmission data from the FIFO buffer, and "1" is set for the CSCRN:STS[5] bit.</p> <p><Notes></p> <ul style="list-style-type: none"> - Because this register is used during the DMA operation, set "0" for this bit during the IO mode. - Set the ST,EA bit before operating the DMA mode.

<Note>

"n" in Buffer Interface Register n (BUFARn) indicates the channel number of the FIFO buffer. For example, if the area for the channel 0 of the FIFO buffer is set, the are should be set in the FIFO buffer channel 0 address range register.

60.3.13 Media Stereo Data Setting Register (MSTD)

Media Stereo Data Setting Register (MSTD) is a register to set whether data to be written in each channel of MediaLB is stereo or monaural.

■ Media Stereo Data Setting Register (MSTD)

Figure 60.3-16 displays the bit configuration of the Media Stereo Data Setting Register (MSTD), and Table 60.3-17 displays the function of each bit.

Figure 60.3-16 Bit Configuration of the Media Stereo Data Setting Register (MSTD)

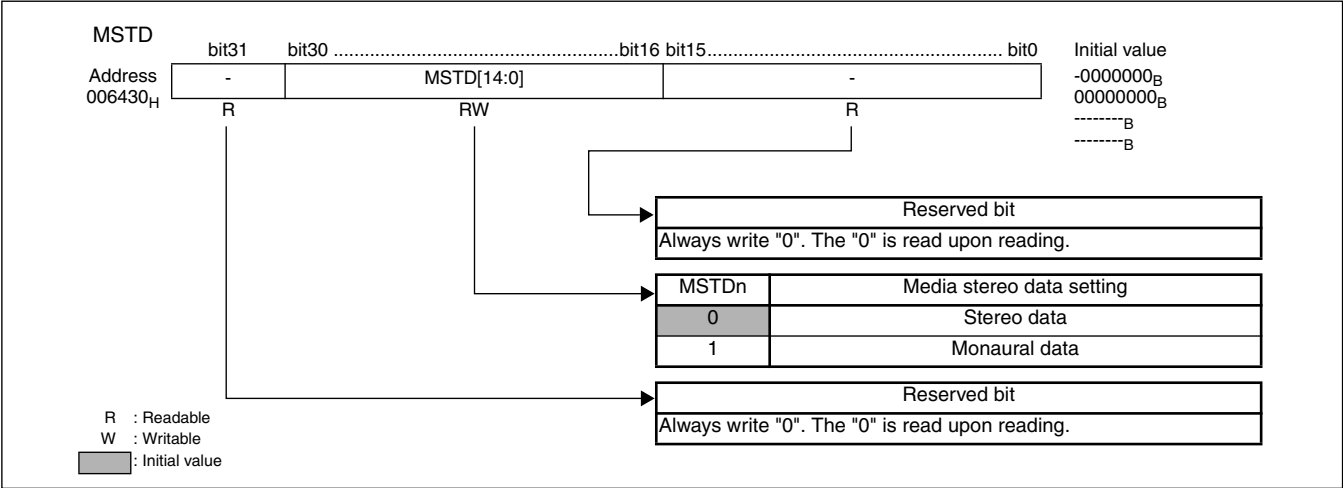


Table 60.3-17 Functional Descriptions on the Media Stereo Data Setting Register (MSTD) Bits

Bit Name		Function
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30- bit16	MSTD[14:0]: Monaural	<p>Sets whether transmission data written in each channel of MediaLB or FIFO buffer is stereo or monaural.</p> <p>MSTD[0]: Logical channel 0 setting bit MSTD[1]: Logical channel 1 setting bit : MSTD[14]: Logical channel 14 setting bit</p> <p>For this bit Set to "0": stereo (16 bit data + 16 bit data) Set to "1": Monaural (16 bit data + 16'h00)</p> <p><Notes></p> <ul style="list-style-type: none"> - During the IO mode, CNBCRN:TDB[31:0] is a target for the stereo and monaural modes. During the DMA mode, BUFNDTR:BUF[31:0] is a target for the stereo and monaural modes. - During the DMA mode, if the monaural mode is used data is written in CNBCRN, set "0" for this register before writing data in CNBCRN. - If the stereo mode (MSTD:MSTDn="0") is set, CNBCRN:TDB[31:0] and BUFNDTR:BUF[31:0] should be written by 32 bits. If the monaural mode (MSTD:MSTDn="1") is set, CNBCRN:TDB[31:0] should be written by 16 bits. If address 1 is "L", "0" is written in CNBCRN:TDB[31:16] and BUFNDTR:BUF[31:16], and 16 bit transmission data is written in CNBCRN:TDB[15:0] and BUFNDTR:BUF[15:0]. If address 1 is "H", 16 bit transmission data is written in CNBCRN:TDB[31:16] and BUFNDTR:BUF[31:16], and "0" is written in CNBCRN:TDB[15:0] and BUFNDTR:BUF[15:0].
bit15- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.

60.3.14 Access Select Register (ASLR)

Access Select Register (ASLR) is a register to select the target for data communication. For FIFO buffers 8 to 14, data transfer is performed between MediaLB and software.

■ Access Select Register (ASLR)

Figure 60.3-17 displays the bit configuration of the Access Select Register, and Table 60.3-18 displays the function of each bit.

Figure 60.3-17 Bit Configuration of the Access Select Register (ASLR)

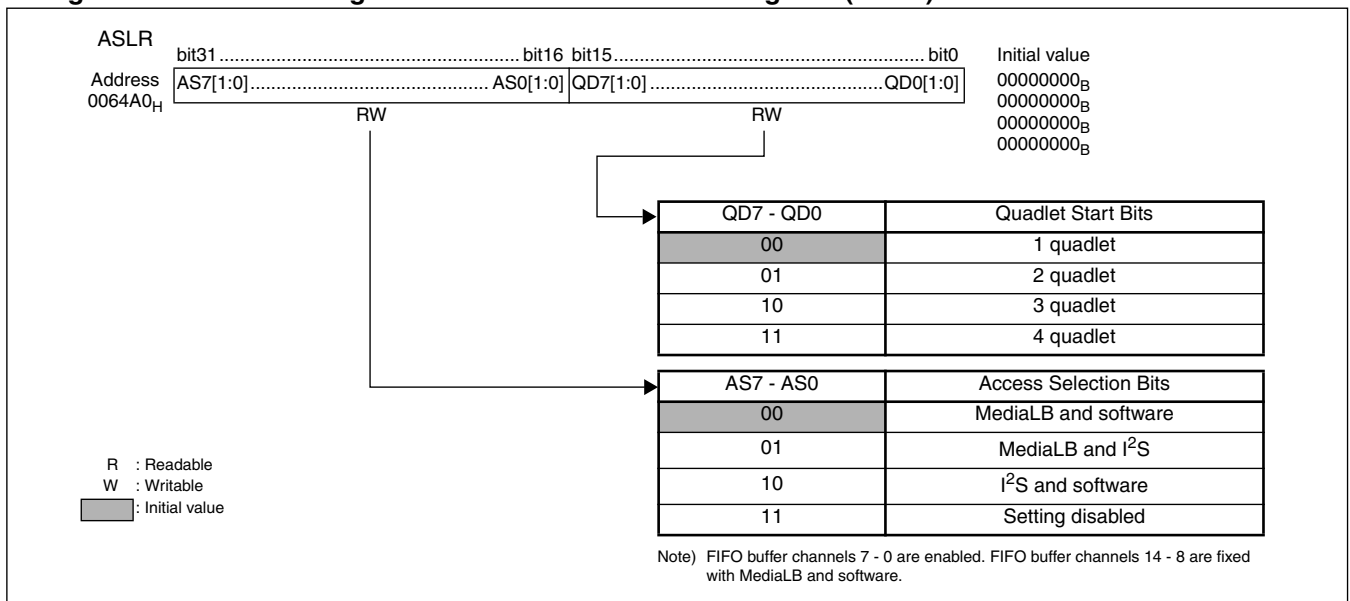


Table 60.3-18 Functional Descriptions on the Access Select Register Bits

Bit Name		Function
bit31-bit16	AS7[1:0] to AS0[1:0]: Access Selection Bits	<p>Selects from where the channel buffer of the FIFO buffer (buffer separated for each logical channel of MediaLB) is accessed. Channel buffers 7 to 0 are targeted for access. Channel buffers 8 to 14 can be accessed only from MediaLB and software.</p> <p>Set to "00": Access from MediaLB and software to the data register is enabled.</p> <p>Set to "01": Access from MediaLB and I²S to the data register is enabled.</p> <p>Set to "10": Access from I²S and software to the data register is enabled.</p> <p>It is prohibited to set "11".</p> <p>If the AS0[1:0] is set to "01", data for channel buffer 0 can be communicated with ch.0 of I²S, and if the AS7[1:0] bit is set to "01", data for channel buffer 7 can be communicated with ch.7 of I²S.</p> <p><Notes></p> <ul style="list-style-type: none"> - Set the channel buffer before setting "1" for this bit. - Setting this bit to "11" is prohibited.
bit14-bit0	QD7[1:0] to QD0[1:0]: Quadlet Start Bits	<p>Starts data transfer between I²S and MediaLB (ASn[1:0]="01") after storing the set number of data into the buffer.</p> <p>Set to "00": Starts data transfer if valid data for 1 quadlet is stored in the FIFO buffer channel after the BUFDCR:I2SRQE bit is changed from "0" to "1".</p> <p>Set to "01": Starts data transfer if valid data for 2 quadlets is stored in the FIFO buffer channel after the BUFDCR:I2SRQE bit is changed from "0" to "1".</p> <p>Set to "10": Starts data transfer if valid data for 3 quadlets is stored in the FIFO buffer channel after the BUFDCR:I2SRQE bit is changed from "0" to "1".</p> <p>Set to "11": Starts data transfer if valid data for 4 quadlets is stored in the FIFO buffer channel after the BUFDCR:I2SRQE bit is changed from "0" to "1".</p> <p>The setting for this bit is ignored once data transfer is started.</p> <p><Notes></p> <ul style="list-style-type: none"> - Valid only for the ASn[1:0]="01" setting. - If software reset is executed, data transfer is started when valid data is stored in the FIFO buffer channel in accordance with this bit setting.

60.3.15 FIFO buffer Direction Control Register (BUFDCR)

FIFO Buffer Direction Control Register sets the access direction from MediaLB to the FIFO buffer.

■ FIFO Buffer Direction Control Register (BUFDCR)

Figure 60.3-18 displays the bit configuration of the FIFO Buffer Direction Control Register, and Table 60.3-19 displays the function of each bit.

Figure 60.3-18 FIFO Bit Configuration of the FIFO Buffer Direction Control Register (BUFDCR)

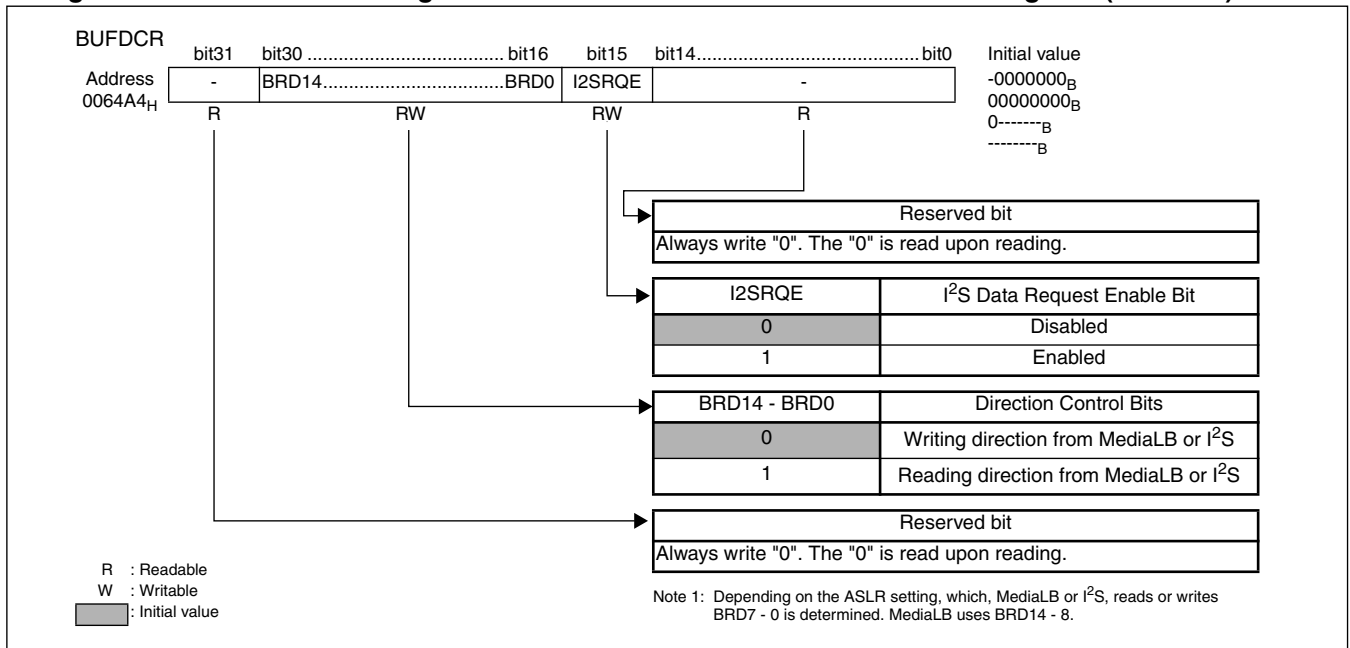


Table 60.3-19 Functional Descriptions on the FIFO Buffer Direction Control Register Bits

Bit Name		Function															
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.															
bit30- bit16	BRD14 to BRD0: Direction Control Bits	<p>Determines the read and write direction to the FIFO buffer. Set to "0": Write direction from MediaLB to the FIFO buffer Set to "1": Read direction from MediaLB to the FIFO buffer However, BRD7 to 0 will become as follows depending on the setting for Access Select Register (ASLR):</p> <table border="1"> <thead> <tr> <th>ASn[1:0]</th><th>BRDn</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>0x</td><td>0</td><td>Write direction from MediaLB to the FIFO buffer</td></tr> <tr> <td>10</td><td>0</td><td>Write direction from I²S to the FIFO buffer</td></tr> <tr> <td>0x</td><td>1</td><td>Read direction from MediaLB to the FIFO buffer</td></tr> <tr> <td>10</td><td>1</td><td>Read direction from I²S to the FIFO buffer</td></tr> </tbody> </table> <p>n=0,1,2,...,7 x=Undefined <Note> It is prohibited to change BRDn when valid data exists.</p>	ASn[1:0]	BRDn	Meaning	0x	0	Write direction from MediaLB to the FIFO buffer	10	0	Write direction from I ² S to the FIFO buffer	0x	1	Read direction from MediaLB to the FIFO buffer	10	1	Read direction from I ² S to the FIFO buffer
ASn[1:0]	BRDn	Meaning															
0x	0	Write direction from MediaLB to the FIFO buffer															
10	0	Write direction from I ² S to the FIFO buffer															
0x	1	Read direction from MediaLB to the FIFO buffer															
10	1	Read direction from I ² S to the FIFO buffer															
bit15	I ² SRQE: I ² S Data Request Enable Bit	<p>Sets whether data should be transferred against the data request from I²S. When ASn[1:0]="01", "10" is set, the following operations are performed by this bit: Set to "0": Transfer is not performed even if data is requested from I²S. Set to "1": Transfer is performed if data is requested from I²S. <Note> If this bit is set to "0" during data transfer, the "0" setting for this bit becomes valid after the completion of data transfer.</p>															
bit14- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.															

60.3.16 FIFO Buffer Interrupt Enable Register (BUFIER)

FIFO Buffer Interrupt Enable Register (BUFIER) sets whether interrupt should be generated when "1" is set for the FIFO buffer status register (BUFSR).

■ FIFO Buffer Interrupt Enable Register (BUFIER)

Figure 60.3-19 displays the bit configuration of the FIFO Buffer Interrupt Enable Register (BUFIER), and Table 60.3-20 displays the function of each bit.

Figure 60.3-19 Bit Configuration of the Buffer Interrupt Enable Register (BUFIER)

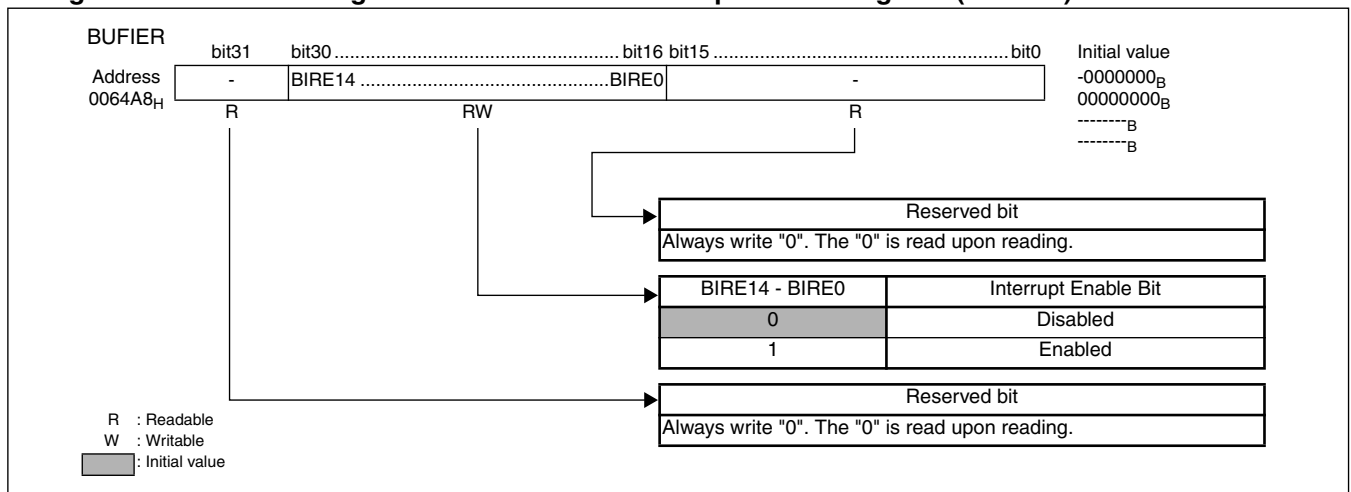


Table 60.3-20 Functional Descriptions on the FIFO Buffer Interrupt Enable Register (BUFIER) Bits

Bit Name		Function
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30- bit16	BIRE14 to BIRE0: Interrupt Enable Bit	Enables interrupt. When "1" is set in the FIFO buffer status register (BUFSR) bit, the setting becomes as follows depending on the corresponding bit setting: Set to "0": Interrupt does not occur. Set to "1": Interrupt occurs.
bit15- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.

60.3.17 FIFO Buffer Status Register (BUFSR)

FIFO Buffer Status Register (BUFSR) indicates whether the setting is below the FIFO buffer threshold.

■ FIFO Buffer Status Register (BUFSR)

Figure 60.3-20 displays the bit configuration of the FIFO Buffer Status Register, and Table 60.3-21 displays the function of each bit.

Figure 60.3-20 Bit Configuration of the FIFO Buffer Status Register (BUFSR)

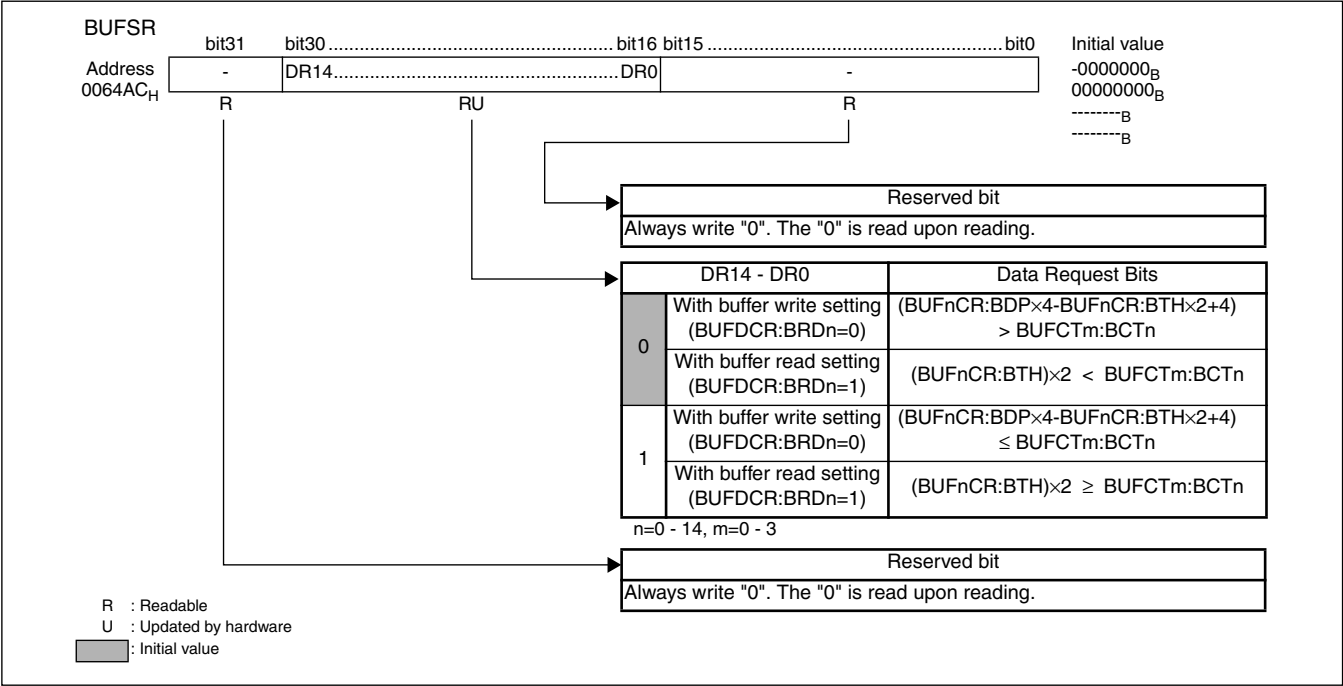


Table 60.3-21 Functional Descriptions on the FIFO Buffer Status Register (BUFSR) Bits

Bit Name		Function
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30- bit16	DR14 to DR0: Data Request Bits	<p>Indicates the data request. Conditions for becoming "1" depend on the data direction setting (BUFDCR:BRDn).</p> <ul style="list-style-type: none"> BUFDCR:BRDn="0" (Write direction from MediaLB) Conditions for becoming "1": Data is written from MediaLB and valid data reaches above (BUFnCR:BDP¥4-BUFnCR:BTH¥2+4). Conditions for becoming "0": Data is written from MediaLB and valid data reaches below (BUFnCR:BDP¥4-BUFnCR:BTH¥2+4). BUFDCR:BRDn="1" (read direction from MediaLB) Conditions for becoming "1": Valid data is below (BUFnCR:BTH)¥2. Conditions for becoming "0": Valid data is beyond (BUFnCR:BTH)¥2. <p><Note> The setting becomes as follows by software reset, depending on the BUFDCR:BRDn setting: BUFDCR:BRDn="0"->DRn="0" BUFDCR:BRDn="1"->DRn="1"</p>
bit15- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.

60.3.18 FIFO Buffer Error Register (BUFER)

FIFO Buffer Error Register (BUFER) displays an error in the event of overrun and incorrect direction setting.

■ Bit Configuration of the FIFO Buffer Error Register (BUFER).

Figure 60.3-21 displays the bit configuration of the FIFO Buffer Error Register (BUFER), and Table 60.3-22 displays the function of each bit.

Figure 60.3-21 Bit Configuration of the FIFO Buffer Error Register (BUFER)

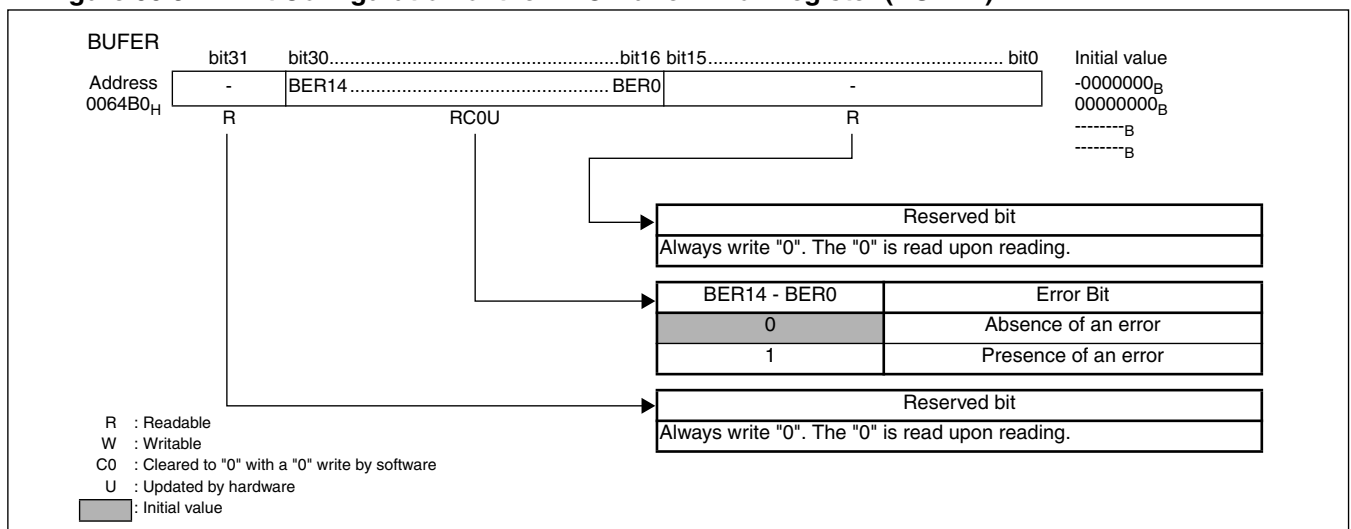


Table 60.3-22 Functional Descriptions on the FIFO Buffer Error Register (BUFER) Bits

Bit Name		Function
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30- bit16	BER14 to BER0: Error Bit	Indicates that an error occurred. This bit is set to "1" in the following conditions. <ul style="list-style-type: none"> - If accessed from Media LB, I²S differently from the BUFDCR:BRDn setting - If overrun occurred When "0" is written in this bit, the bit is cleared to "0". Invalid if "1" is written. <Notes> <ul style="list-style-type: none"> - This bit is cleared to "0" by the software reset. - "1" is read out upon the read modify write command. - This bit is not set to "1" even if the access direction from the software is incorrect. At that time, the write access is disabled, and readout data during read access is read out.
bit15- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.

60.3.19 FIFO Buffer Reset Register (BUFRST)

FIFO Buffer Reset Register (BUFRST) initializes the address pointer, buffer counter, status information and error information for the buffer (channel buffer) separated according to the logical channel of MediaLB.

■ Bit Configuration of the FIFO Buffer Reset Register (BUFRST)

Figure 60.3-22 displays the bit configuration of the FIFO Buffer Reset Register (BUFRST), and Table 60.3-23 displays the function of each bit.

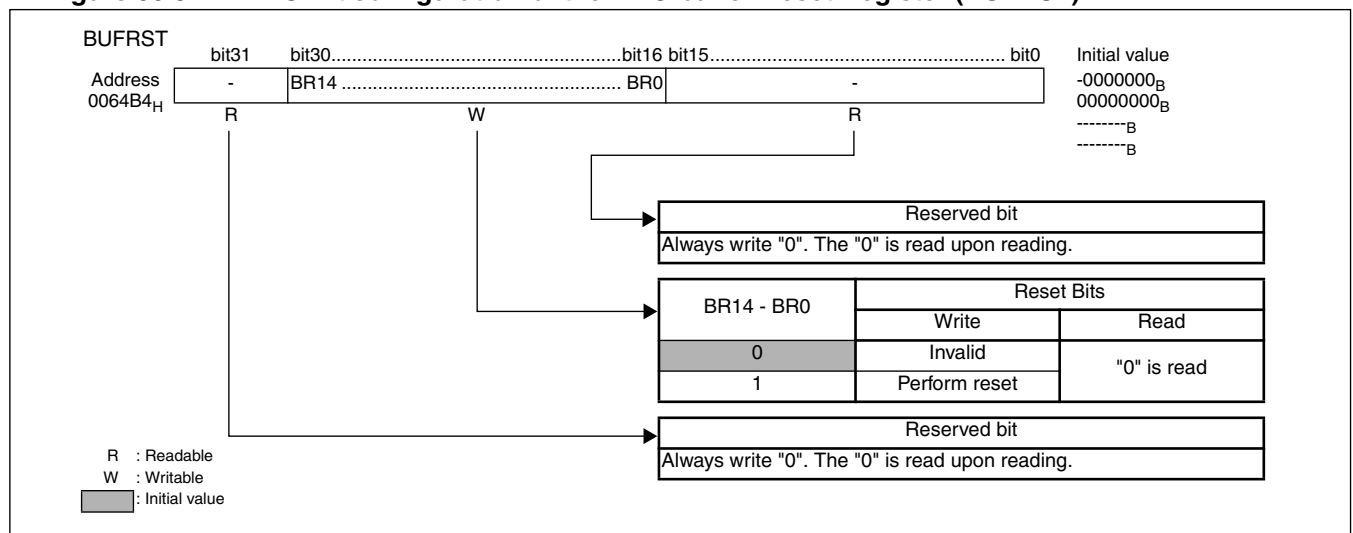
Figure 60.3-22 FIFO Bit configuration of the FIFO buffer Reset Register (BUFRST)

Table 60.3-23 Functional Descriptions on the FIFO Buffer Reset Register (BUFRST) Bits

Bit Name		Function
bit31	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30- bit16	BR14 to BR0: Reset Bits	Performs software reset. When "1" is written in this bit, the address pointer, buffer counter, status information, and error information for the FIFO buffer channel (buffer separated for each MediaLB logical channel) are initialized. In this bit, "0" is read upon reading. Writing of "0" is invalid.
bit15- bit0	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.

60.3.20 FIFO Buffer Count Register m (BUFCTm)[m=0,1,2, • • • ,7]

FIFO Buffer Count Register m (BUFCTm) displays the number of valid data in the channel buffer (FIFO buffer separated for each logical channel of MediaLB).

■ Bit Configuration of the FIFO Buffer Count Register m (BUFCTm)

Figure 60.3-23 displays the bit configuration of the FIFO Buffer Count Register m (BUFCTm), and Table 60.3-24 displays the function of each bit.

Figure 60.3-23 FIFO Bit Configuration of the FIFO Buffer Count Register m (BUFCTm)

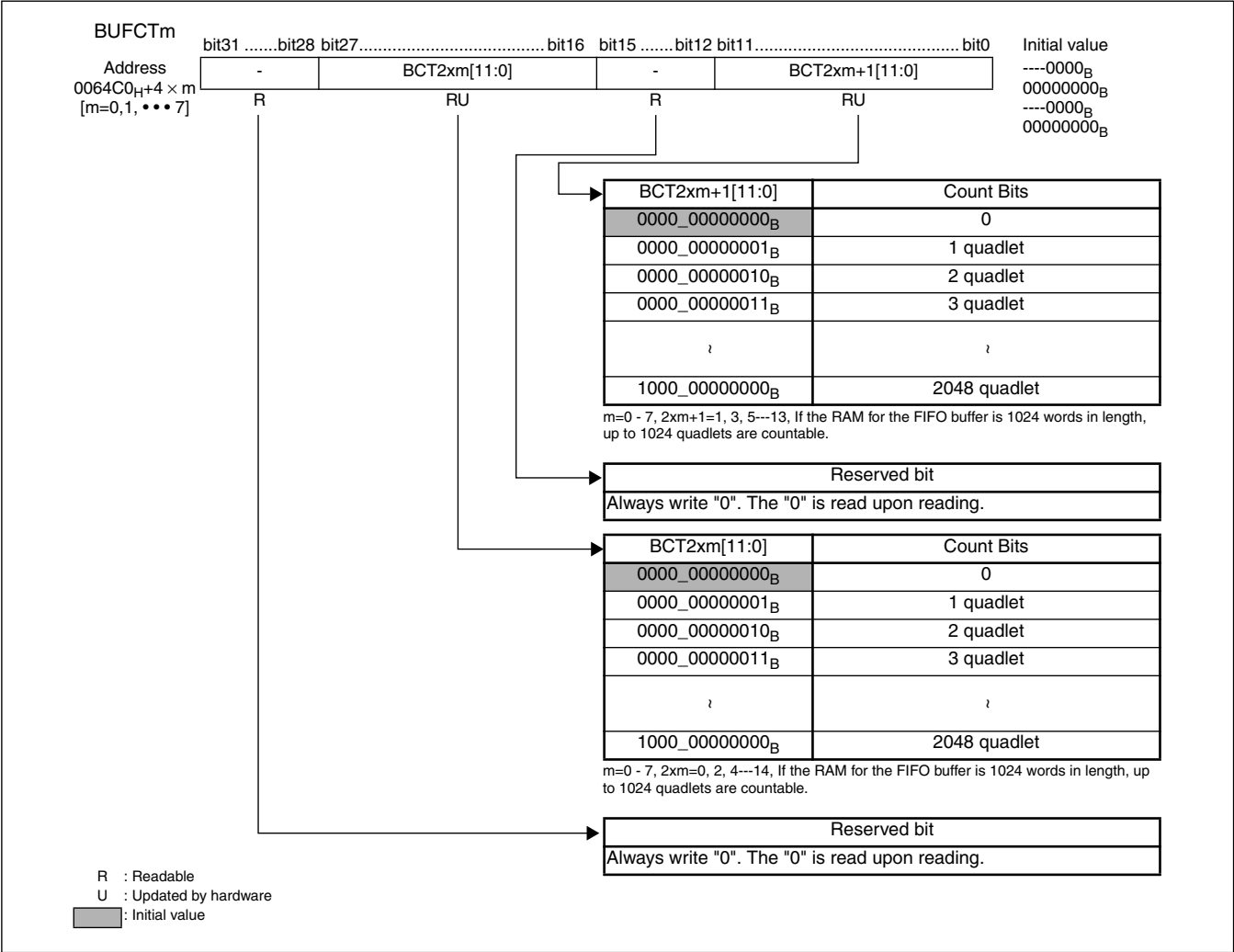


Table 60.3-24 Functional Descriptions on the FIFO buffer Count Register m (BUFCTm) Bits

Bit Name		Function
bit31-bit28, bit15-bit12	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit27-bit16, bit11-bit0	BCT2xm, BCT2xm+1: Count Bits	Indicates the number of valid data. Controls in units of 4 byte data (1 quadlet) <Notes> <ul style="list-style-type: none"> - The bit is cleared to "0" by the software reset. - During the read direction from MediaLB (BUFDCR:BRDn="1"), 0 is returned to MediaLB if there is no valid data in the channel buffer (FIFO buffer separated for each logical channel of MediaLB). - When read from I²S in the write direction from MediaLB, data request from I²S is not accepted if there is no effective data in the channel buffer. At that time, "0" is output as serial data from I²S and the ERR bit of I²S becomes "1".

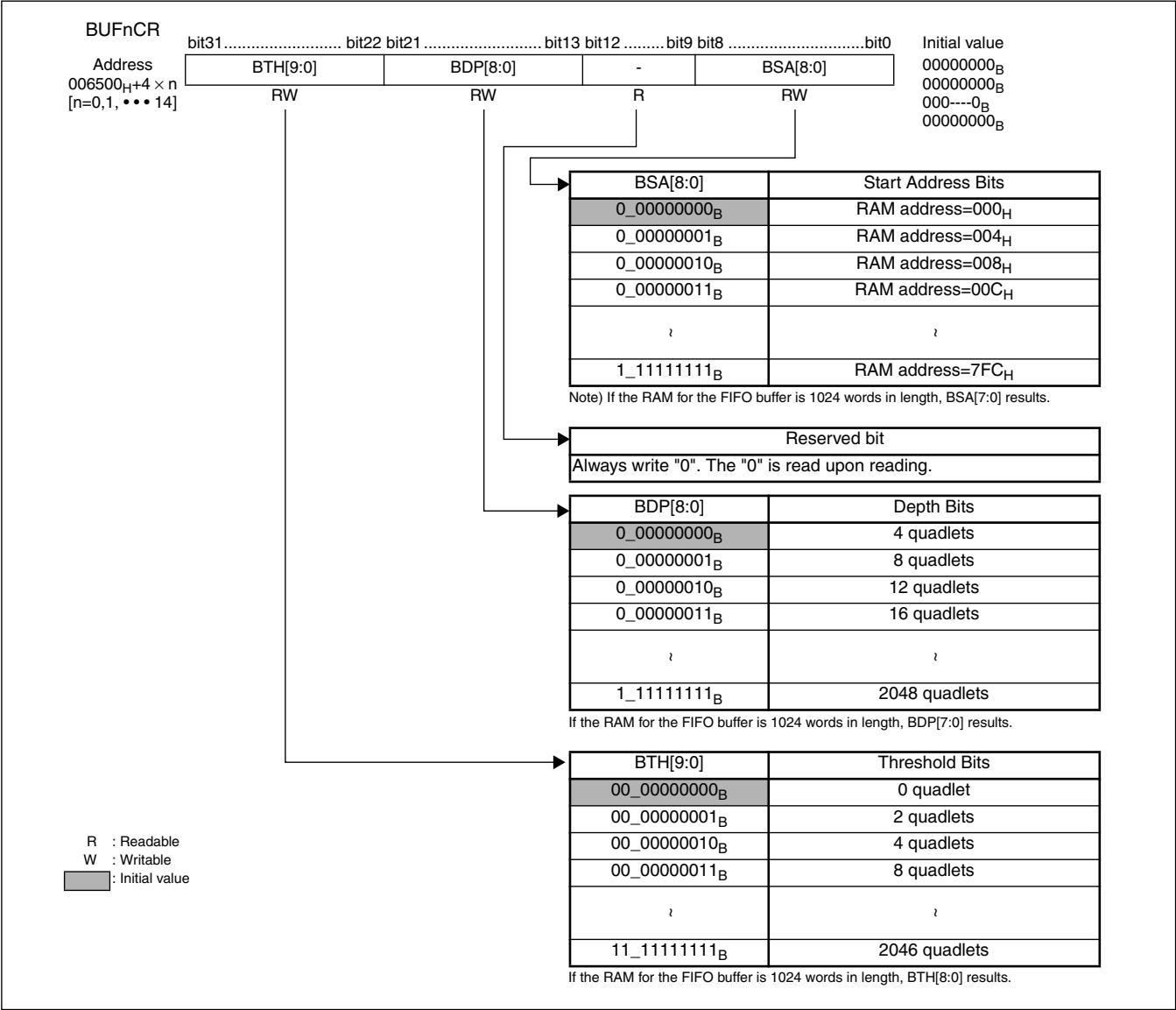
60.3.21 FIFO buffer n Control Register (BUFnCR) [n=0,1,2, • • • ,14]

FIFO Buffer Control Register (BUFnCR) sets the start address, depth, and number of quadlets in which interrupt occurs for the channel buffer (FIFO buffer separated for each logical channel of MediaLB).

■ Bit Configuration of the FIFO Buffer n Control Register (BUFnCR)

Figure 60.3-24 displays the bit configuration of the FIFO Buffer n Control Register (BUFnCR), and Table 60.3-25 displays the function of each bit.

Figure 60.3-24 FIFO Bit Configuration of the FIFO Buffer n Control Register (BUFnCR)



MB91460 Series

Table 60.3-25 Functional Descriptions on the FIFO Buffer n Control Register (BUFnCR) Bits

Bit Name		Function
bit31-bit22	BTH: Threshold Bits (Quadlet/2)	<p>Sets a boundary where a data request is made. A threshold is set in units of 2 quadlets.</p> <p>BTH=000_H -> 0 quadlet BTH=001_H -> 2 quadlets BTH=002_H -> 4 quadlets BTH=3fe_H -> 2044 quadlets BTH=3ff_H -> 2046 quadlets</p> <p>When the following conditions are met, each bit in FIFO Buffer Status Register is set to "1".</p> <ul style="list-style-type: none"> - When BUFDnCR:BRDn="0" BUFnCR:BDPx4-BUFnCR:BTHx2+4 ≤ BUFDnCR:BCTn - When BUFDnCR:BRDn="1" (BUFnCR:BTH)x2 ≤ BUFDnCR:BCTn <p><Notes></p> <ul style="list-style-type: none"> - If FIFO buffer RAM is 1024 words in length, up to 1ff_H can be set. - Set this bit to a value not more than that of BDP.
bit21-bit13	BDP: Depth Bits ((Quadlet /4)-1)	<p>Sets a RAM area. A length is set in units of 4 quadlets.</p> <p>BDP=000_H -> 4 quadlets BDP=001_H -> 8 quadlets BDP=002_H -> 12 quadlets BDP=1fe_H -> 2044 quadlets BDP=1ff_H -> 2048 quadlets</p> <p><Notes></p> <ul style="list-style-type: none"> - If FIFO buffer RAM is 1024 words in length, up to ff_H can be set. - Set a RAM area so that it does not overlap with the RAM area for a channel buffer used. - Set LCBCRn:BD for MediaLB and this bit so that the following conditions are met. LCBCRn:BD ≤ BUFnCR:BDP
bit12-bit9	rsvd: Reserved	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit8-bit0	BSA: Start Address Bits (Quadlets/4)	<p>Sets a RAM start address. The actual RAM address is BSA 4 quadlets.</p> <p><Note></p> <p>Set the start addresses of channel buffers used so that they do not overlap with each other.</p>

60.3.22 FIFO Buffer n Data Register (BUFnDTR)[n=0,1,2, ...,14]

FIFO Buffer n Data Register (BUFnDTR) is a register used to transmit and receive data to and from a channel buffer, a FIFO buffer that is divided into areas each of which corresponds to a logical channel of MediaLB. Be sure to access in 32 bits.

■ Bit Configuration for FIFO Buffer n Data Register (BUFnDTR)

Figure 60.3-25 shows the bit configuration for FIFO Buffer n Data Register (BUFnDTR), and Table 60.3-26 provides description of each bit's function.

Figure 60.3-25 Bit Configuration for FIFO Buffer n Data Register (BUFnDTR)

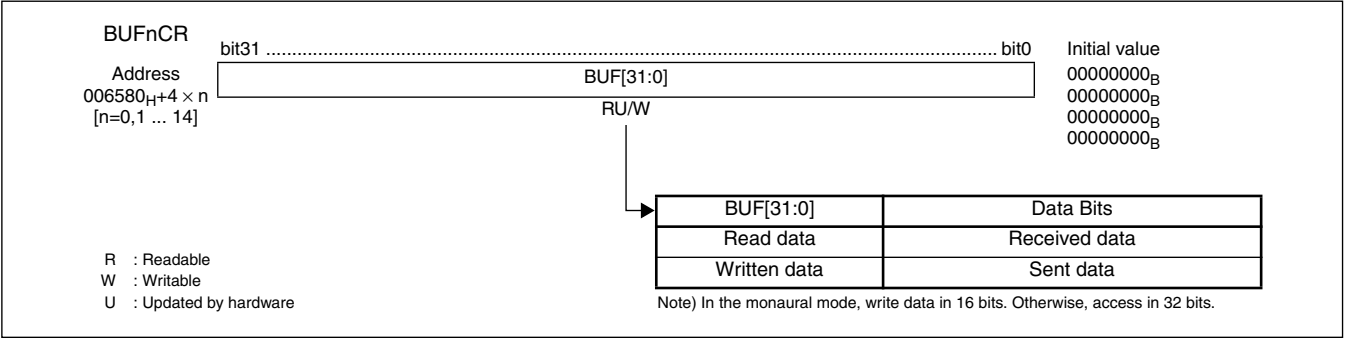


Table 60.3-26 Functional Description of Each Bit in FIFO Buffer n Data Register (BUFnDTR)

Bit Name		Function
bit31-bit0	BUF: Data Bits	<p>Transmits and receives data to and from MediaLB or I²S. Direction Control Bit (BUFDCR:BRDn) is used to determine whether reading or writing is enabled.</p> <ul style="list-style-type: none">- If BUFDCR:BRDn="0" Reading of received data is enabled.- If BUFDCR:BRDn="1" Writing of transmitted data is enabled. <p><Notes></p> <ul style="list-style-type: none">- In the monaural mode, write data in 16 bits. Otherwise, be sure to access data in 32 bits.- Data is transferred twice only when it is through Channel 8 or 9. If you want to transfer data twice, set the DBL bit for I²S to "1" to write the transmitted data to Channel 8 or 9.

60.4. Description of MediaLB Behavior

MediaLB supports two data transfer modes: IO and DMA. It also supports the loopback test mode for debugging.

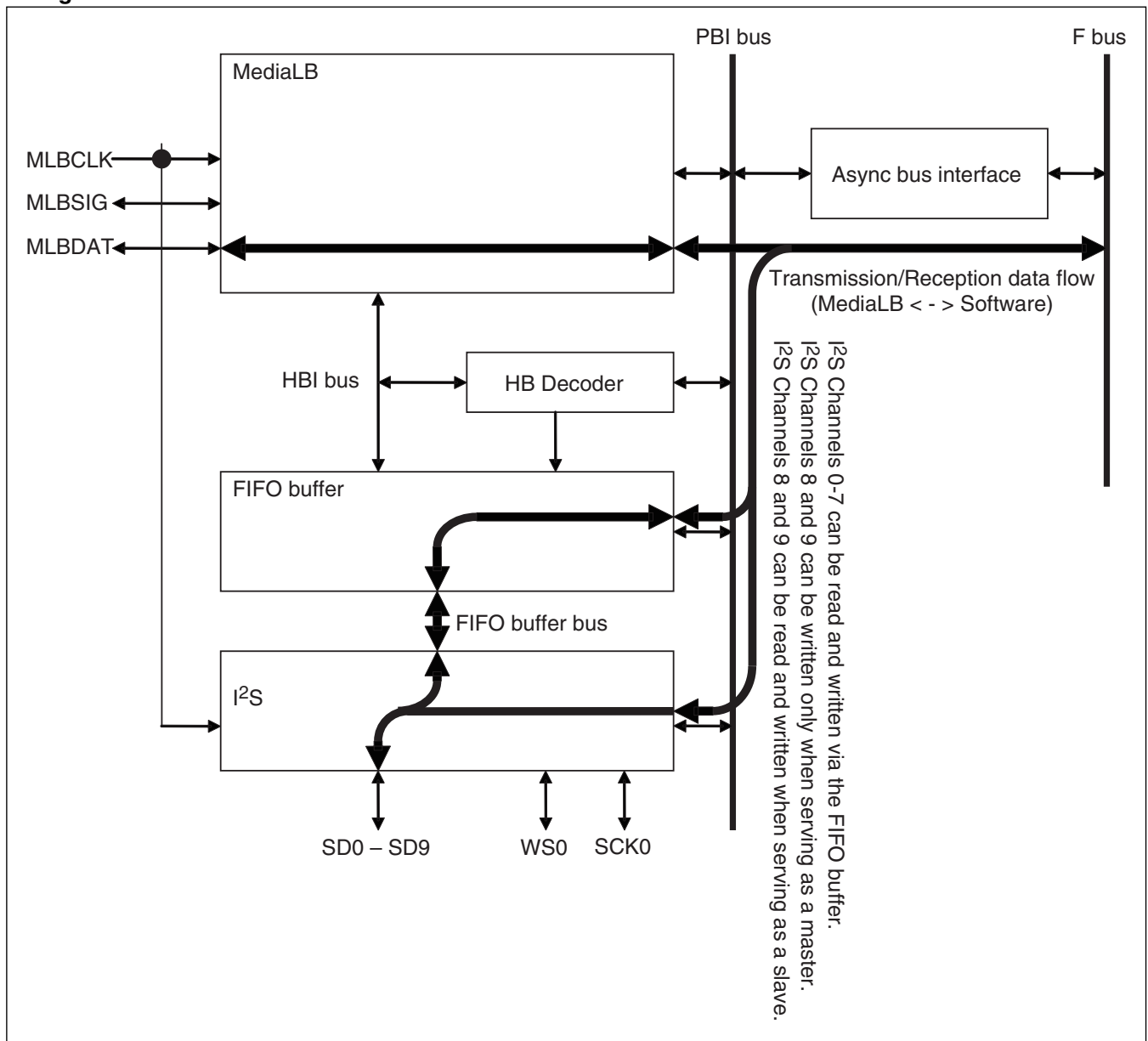
This section describes MediaLB's IO and DMA modes (ping-pong and circular buffering), the local channel buffer and the behavior of the FIFO buffer.

60.4.1 IO mode

In the IO mode, MediaLB transmits and receives data to and from software via the PBI bus.

The IO mode is a mode in which MediaLB reads or writes the data that has been transmitted or received to and from software from or to the local channel buffer.

Transmission/Reception data flow

MB91460 Series**Figure 60.4-1 Data Flow When MediaLB is in the IO Mode**

When MediaLB is used in the IO mode, data is transmitted and received via the PBI bus between software and MediaLB or between software and I²S.

Channels 0 - 7 of I²S can transmit and receive data by using the FIFO buffer. Channels 8 and 9 of I²S can only transmit data when serving as a master. They can transmit and receive data when serving as a slave.

When the FIFO buffer is used, Channels 0 - 7 of I²S are connected to Channels 0 - 7 of the FIFO buffer. So, it is necessary to match the reading/writing direction for I²S with the transmit/write direction for the FIFO buffer. If they do not agree, the Error Bit (BUFER:BERn) for the FIFO Buffer is set to "1". The following paragraph shows a setting example:

Example) Using I²S Channel 0 for transmitting data

I2SSCR0:TR="1" (I²S transmits data)

ASLR:AS0[1:0]= "10" (Access between I²S and the FIFO buffer)

BUFDCR:BRD0="1" (Reading direction from I²S)

BUFDCR:I2SRQE="1" (Enables access from the I²S.)

Example) Using I²S Channel 1 for receiving data

I2SSCR1:TR="0" (I²S receives data)

ASLR:AS1[1:0]="10" (Access between I²S and the FIFO buffer)

BUFDCR:BRD1="0" (Writing direction from I²S)

BUFDCR:I2SRQE="1" (Enables access from the I²S.)

■ Behavior in IO Mode

When putting MediaLB in the IO mode, the CCBCRn and CNBCRn registers are used as a data receive buffer and a data transmit buffer, respectively. When received data is read from the data receive buffer, the received data is read from the local channel buffer in MediaLB. When transmit data is written in the data transmit buffer, the transmitted data is written in the local channel buffer in MediaLB.

<Notes>

- Make sure to set the start address, depth and threshold of the local channel buffer before setting the channel enable (CECRn:CE) to "1".
- Make sure to set the logical channel to be used to either IO mode or DMA mode.

(1) Transmitting from MediaLB

If the TR bit in Channel n Entry Configuration Register (CECRn) is set to "1", it indicates a transmitting direction.

Writing the transmit data to the CNBCRn register causes the BM bit in the CSCRn register to be set to "0". The data continues to be written until the amount of data written exceeds the amount of data designated by the TH bit in LCBCRn register, at which time the Send Service Request Bit (STS[3]) in the CSCRn register is changed from "1" to "0". When Local Channel n Buffer is full of valid send buffers, the BF bit in the CSCRn register is set to "1".

When the channel address in a MediaLB frame agrees with the channel address of the logical channel n, MediaLB issues a command to the MLBSIG signal, and at the same time, transmits data in response to the MLBDAT signal. When the amount of data left becomes not more than the amount designated by the TH bit in Local Channel n Buffer while data is being transmitted in response to the MLBDAT signal, the Send Service Request Bit is set to "1" again, triggering an interrupt unless the interrupt is masked. Transmit data is further read from the Local Channel n Buffer until the transmitted data is no longer left, at which time the EMPTY bit in the CSCRn register is set to "1".

For async or control channels, AsyncBreak (26_H) and ControlBreak(36_H) commands are issued under the following conditions:

- The GB bit in the CSCRn register is set to "1". The channel address of the channel agrees, ReceiverBusy(10_H) is received as RxSTATUS and then the channel address agrees again.

<Note>

Set the upper 16 bits in a packet to the packet length (in bytes).

(2) Receiving from MediaLB

When setting the TR bit in Channel n Entry Configuration Register (CECRn) to "0", the channel is used as a receiving channel.

If the channel address in a MediaLB frame agrees with the channel address of the logical channel n, the following response occurs, depending on the channel type unless a protocol error occurs.

- If the BF bit in the CSCRn register is set to "0" for an async or control channel, MediaLB returns ReceiverReady(00_H) as RxSTATUS.
- If the BF bit in the CSCRn register is set to "1" for an async or control channel, MediaLB returns ReceiverReady(10_H) as RxSTATUS.
- If the GB bit in the CSCRn register is set to "1" for an async or control channel, MediaLB returns ReceiverReady(70_H) as RxSTATUS.
- For a sync channel, MediaLB returns ReceiverReady(00_H) as RxSTATUS.

After transmitting a response as shown above, MediaLB receives data.

Depending on the relationship between the value of the TH bit in the LCBCRn register or the PL bit in CECRn register and the amount of received data, Receive Service Request Bit (STS[2]) in the CSCRn register is changed to "1", triggering an interrupt unless the interrupt is masked. MediaLB reads received data from the CCBCRn register until the BM bit in the CSCRn register is changed to "1". When the BM bit in the CSCRn register is changed to "1", it indicates that there is no longer received data left in Local Channel n Buffer.

<Notes>

- The upper 16 bits in a packet indicates the packet length (in bytes).
 - Depending on the setting of the TH bit in the LCBCRn register or the PL bit in CECRn register, received data is left in Local Channel n Buffer, without triggering an interrupt.
-

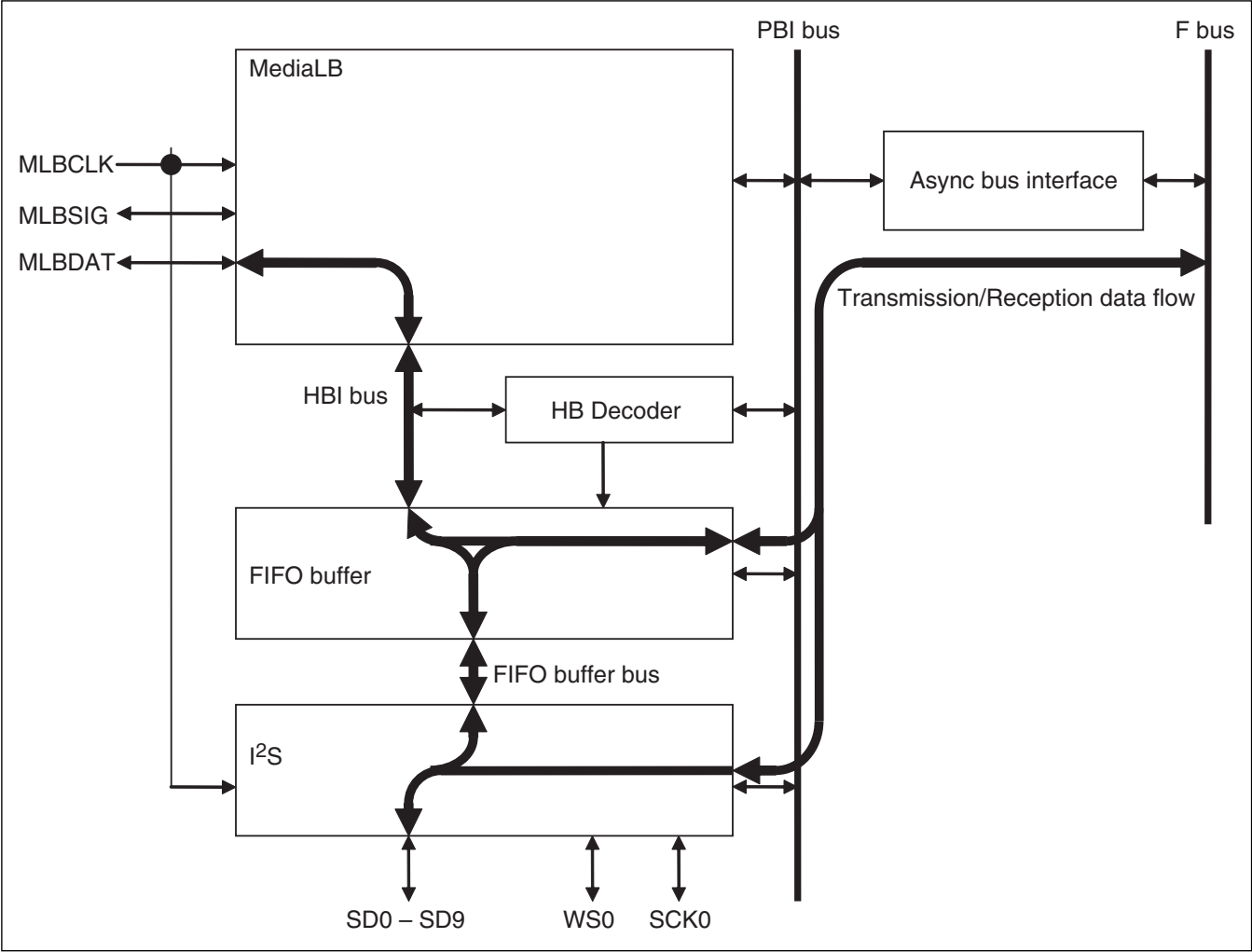
60.4.2 DMA mode

In the DMA mode, MediaLB transmits and receives data from and to the FIFO buffer and I²S via the HBI bus.

The DMA mode is a mode in which MediaLB accesses the FIFO buffer via the HBI bus. The FIFO buffer contains transmitted and received data, which can be read by software, I²S and MediaLB.

■ Transmission/Reception data flow

Figure 60.4-2 Data Flow When MediaLB is in the DMA Mode



When MediaLB is used in the DMA mode, data is transmitted and received via the FIFO buffer. When used in the DMA mode, MediaLB functions as a master of the HBI bus, reading and writing data from and to the FIFO buffer.

When MediaLB accesses the HBI bus, Buffer Current Address BCA[15:0] in Channel n Current Buffer Configuration Register (CCBCRn) for MediaLB is output as the address of the HBI bus, and MediaLB compares it with FIFO Buffer Channel n Address Range Register (BUFARn) in the HB Decoder block to determine which channel in the FIFO buffer has that address, and then activates the Channel Selection signal from HB Decoder to access the selected channel in the FIFO buffer.

In the DMA mode, two types of buffering is supported: Ping-pong and circular buffering. It is recommended that circular buffering should be used when using synchronous data between MediaLB and I²S.

<Notes>

- Make sure to set the start address, depth and threshold of the local channel buffer before setting the channel enable (CECRn:CE) to "1".
 - Before using the FIFO buffer, set FIFO Buffer n Control Register (BUFnCR) and FIFO Buffer Channel n Address Range Register (BUFARn).
 - Make sure to set the logical channel to be used to either IO mode or DMA mode.
-

■ Ping-pong Buffering

In the ping-pong buffering, the start and end addresses are set in Next Buffer Configuration Register to transfer data between the addresses.

Before performing ping-pong buffering, make the following settings:

- Setting of the device configuration register (DCCR)
- Setting of the channel n entry configuration register (CECRn)
- Setting of the local channel n buffer configuration register (LCBCRn)
- Setting of the FIFO buffer channel n address range register (BUFARn)
- Setting of the access select register (ASLR)
- Setting of the FIFO buffer direction control register (BUFDCR)
- Setting of the FIFO buffer n control register (BUFnCR)

(1) Transmitting Data via Async or Control Channel

When transmitting a MediaLB frame, it is necessary to match the reading/writing direction for MediaLB with the transmit/write direction for the FIFO buffer by setting the TR bit in Channel n Entry Configuration Register (CECRn) to "1" and BRDn in FIFO Buffer Direction Control Register (BUFDCR) to "1". If there is a mistake in setting, set the host error bit in the channel n status configuration register (CSCRn) of the MediaLB and the error bit in the FIFO buffer error register (BUFER) to "1".

Set the MediaLB according to the following procedure to make ping-pong buffering operate.

(a) Set the address corresponding to the channel address that FIFO buffer uses in the channel n next buffer configuration register (CNBCRn).

- It is necessary to set the BSA bit in Channel n Next Buffer Configuration Register (CNBCRn) to the same value as that of the ST bit in FIFO Buffer Channel n Address Range Register (BUFARn) and the BEA bit to the BSA value plus the number of bytes transmitted. In the setting, make sure to satisfy the condition that CNBCRn:BEA<BUFARn:EA.

Example: When the FIFO buffer uses Channel 1 (BUFAR1:ST[15:0]=2000_H and BUFAR1:EA=2010_H) to transmit 10 byte data via an async channel

- Set CNBCR1:BSA[15:0] to 2000_H and CNBCR1:BEA[15:0] to 2008_H
- Begin with writing the first data piece that contains the number of bytes of data transmitted in

the upper 16 bits (in this case, 000A_H is written in BUF1DTR:BUF[31:16]) and actually transmitted data in the lower 16 bits to the FIFO buffer. Then, continue to write the remaining transmitted data to the buffer.

- (b) Write the transmitted data to the channel used in the FIFO buffer.
 - Continue to write given transmitted data. For an async or control channel, set 16 bits (Bit 31- Bit 16) of the frame transmitted first to the number of bytes in the frame. Transmitted data can be divided into a plurality of frames, but make sure that the last frame is ended with CNBCRn:BEA.
- (c) Write "1" to the RDY bit in Channel n Status Configuration Register (CSCRn).

When the above settings are made, if ping-pong buffering is not operating, RDY bit in the channel n status configuration register (CSCRn) is set to "0" by hardware and the value of the channel n next buffer configuration register (CNBCRn) is copied to the channel n current buffer configuration register (CCBCRn). At that point, STS[3] (current buffer start) bit in the channel n status configuration register (CSCRn) is set to "1" to start ping-pong buffering. When the RDY bit in Channel n Status Configuration Register (CSCRn) is changed to "0", the above setting can be made for the next data to be transmitted. However, unless the FIFO buffer channel has an area available for writing the next transmitted data, do not make the above setting until STS[2] in the Channel n Status Configuration Register (CSCRn) is changed to "1".

Once ping-pong buffering starts, the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is output as the HBI bus's address, and MediaLB selects a FIFO buffer channel that falls within the range of the setting of FIFO Buffer Channel n Address Range Register (BUFARn) in HB Decoder and reads transmitted data from the selected FIFO buffer channel in a fixed length of 32 bits. When read is complete, MediaLB stores the read data in the local channel buffer, increasing the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) by 4. MediaLB continues to transmit data until the value of the BCA bit becomes equal to that of the BFA bit.

When the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is equal to that of the BFA bit, and the RDY bit in Channel n Status Configuration Register (CSCRn) is set to "0", STS[2] in Channel n Status Configuration Register (CSCRn) is changed to "1", and if the RDY bit is set to "1", STS[10] in Channel n Status Configuration Register (CSCRn) is changed to "1" and the RDY bit is changed to "0".

For an async or control channel, the 16 bits of the packet transmitted first contains the number of bytes of data in the packet. If the received channel address agrees, MediaLB outputs an AsyncStart (20_μ) command for async channels or a ControlStart (30_μ) command for control channels, as well as data, at the beginning of frame, and counts the number of bytes it has transmitted. MediaLB continues to output an AsyncContinue(22_μ) command for async channels or a ControlContinue(32_μ) command for control channels until it has transmitted a given number of bytes. MediaLB outputs AsyncEnd(24_μ) for async channels or ControlEnd(34_μ) for control channels at the end of data output to indicate the end of packets. If the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) does not reach that of the BFA bit after sending a given number of bytes, MediaLB outputs a start command as the next package data to start transmitting the next frames when the received channel address agrees.

<Notes>

- Set the upper 16 bits (But 31 - Bit16) of the packet transmitted first to the number of bytes in the packet.
 - Set the BEA bit so that the data transmitted in the final packet is store in the address indicated by the BEA bit in Channel n Next Buffer Configuration Register (CNBCRn).
-

■ Receiving ReceiverBreak(70_H)

When ReceiverBreak(70_H) has been received as a status response, STS[9] or STS[2] in the Channel n Status Configuration Register (CSCRn) is set to "1", and the remaining data in the packet is discarded, with the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) increased by the number of bytes of the data discarded. The ping-pong buffering operation continues if BCA bit in the channel n current buffer has not yet reached BFA bit.

■ Receiving ReceiverBusy(10_H)

If the GB bit in Channel n Status Configuration Register (CSCRn) is set to "0" when ReceiverBusy(10_H) has been received as a status response, the data for which ReceiverBusy(10_H) was received is transmitted again when the channel address agrees next time. For the GB bit in Channel n Status Configuration Register (CSCRn) set to "1", see "Sending a Break".

■ Receiving ReceiverProtocolError(72_H)

If MediaLB has received ReceiverProtocolError(72_H) as a status response, STS[8] or STS[0] in Channel n Status Configuration Register (CSCRn) is set to "1", and the remaining data in the packet is discarded, with the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) increased by the number of bytes of the data discarded. The ping-pong buffering operation continues if BCA bit in the channel n current buffer has not yet reached BFA bit.

■ Transmitting AsyncBreak(26_H) or ControlBreak(36_H)

AsyncBreak(26_H) and ControlBreak(36_H) are supported for async channels and control channels, respectively. When MediaLB has received ReceiverBusy(10_H) as RxStatus with the GB bit in Channel n Status Configuration Register (CSCRn) set to "1", MediaLB sets Channel n Status Configuration Register (CSCRn) to "0" and then issues AsyncBreak(26_H) or ControlBreak(36_H) command when the channel address received next agrees. At this time, the data in the packet is discarded.

<Note>

- Receiving ReceiverBreak(70_H) as RxStatus with the GB bit in Channel n Status Configuration Register (CSCRn) set to "1" does not cause the GB bit to be cleared to "0".

(2) Transmitting Data Via a Synch Channel

The setting method for sync channels is the same as that for async channels, except that, for sync channels, transmitted data is just written to the FIFO buffer, without the need to set the number of frame bytes in the frame transmitted first.

In addition, even when MediaLB has received ReceiverBusy(10_H) or ReceiverBreak(70_H) as RxSTATUS, MediaLB ignores it to continues to transmit data.

(3) Receiving Data Via an Async or Control Channel

When receiving a MediaLB frame, it is necessary to match the reading/writing direction for MediaLB with the sending/writing direction for the FIFO buffer by setting the TR bit in Channel n Entry Configuration Register (CECRn) to "0" and BRDn in FIFO Buffer Direction Control Register (BUFDn) to "0". If there is a mistake in setting, set the host error bit in the channel n status configuration register (CSCRn) of the MediaLB and the error bit in the FIFO buffer error register (BUFER) to "1".

Set the MediaLB according to the following procedure to make ping-pong buffering operate.

- (a) Set the address corresponding to the channel address that FIFO buffer uses in the channel n next buffer configuration register (CNBCRn).

- The BSA bit in Channel n Next Buffer Configuration Register (CNBCRn) is set to the same value as that of the ST bit in Buffer Channel n Address Range Register (BUFARn), and the BEA bit is set to the BSA value plus the number of bytes received.
As how much data will be received is unknown, an estimated value is set.
Set the EA bit in FIFO Buffer Channel n Address Range Register (BUFARn) to a value larger than the number of bytes received.

(b) Write "1" to RDY bit in the channel n status configuration register (CSCRn).

- When the received channel address agrees with the RDY bit not set to "1", MediaLB stores the received data in the local channel buffer unless the buffer is full. When the local channel buffer is full, it outputs ReceiverBusy as RxSTATUS.

When the above settings are made, if ping-pong buffering is not operating, RDY bit in the channel n status configuration register (CSCRn) is set to "0" by hardware and the value of the channel n next buffer configuration register (CNBCRn) is copied to the channel n current buffer configuration register (CCBCRn). At that point, STS[3] (current buffer start) bit in the channel n status configuration register (CSCRn) is set to "1" to start ping-pong buffering. If the RDY bit in Channel n Status Configuration Register (CSCRn) has been changed to "0", the following transfer is enabled.

When the received channel address agrees, MediaLB behaves as follows:

- Stores the received data in the local channel buffer temporarily.
- Outputs the data saved in the local channel buffer to the HBI bus at the same time when the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is output to the HBI bus as an address.
- Selects a FIFO buffer channel that falls within the range of the value of FIFO Buffer Channel n Address Range Register (BUFARn) in HB Decoder.
- Writes the data on the HBI bus to the selected FIFO buffer channel.
- Increases the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) by 4.

When receiving AsyncEnd(24_H) or ControlEnd(34_H) command with the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) exceeding that of the BFA bit, MediaLB terminates the ping-pong buffering. If the RDY bit in Channel n Status Register (CSCRn) is set to "1" when ping-pong buffering is terminated, MediaLB changes the STS[10] bit in Channel n Status Configuration Register (CSCRn) to "1", starting the next transfer, which also changes the STS[3] bit to "1". When the RDY bit in Channel n Status Configuration Register (CSCRn) is set to "0", MediaLB changes STS[2] in Channel n Status Configuration Register (CSCRn) bit to "1".

<Note>

- The first packet contains the number of bytes in the packet. For received data, the number of bytes is contained in the upper 16 bits (Bit 31- Bit 16) of the data piece received first.
-

■ Receiving AsyncBreak(26_H) or ControlBreak(36_H)

When the received channel address agrees, and MediaLB receives AsyncBreak(26_H) for async channels or ControlBreak(36_H) for control channels, it sets the STS[1] bit in Channel n Status Configuration Register (CSCRn) to "1".

At this time, the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is reloaded with the start address from which onwards the packet data is stored, and MediaLB starts to store received data from that address onwards when receiving AsyncStart(20_H) or ControlStart(30_H).

When STS[3] bit is changed to "1" with the STS[1] bit in Channel n Status Configuration Register (CSCRn) set to "1", the STS[1] bit is reset to "0" and the STS[9] bit is changed to "1".

<Notes>

- Outputting ReceiverBusy(10_H), ReceiverProtocolError(72_H), or ReceiverBreak(70_H) in response to AsyncBreak(26_H) or ControlBreak(36_H) command does not cause the STS[1] in Channel n Status Configuration Register (CSCRn) to be set to "1".
 - Receiving AsyncBreak(26_H) or ControlBreak(36_H) causes BCA in Channel n Current Buffer Configuration Register (CCBCRn) to be reloaded, starting to receive data from the beginning. As the FIFO buffer retains the received data before it receives a break, read successful packets from an appropriate channel in the FIFO buffer to perform software reset on that channel.
-

■ Transmitting ReceiverBreak(70_H)

To outputs ReceiverBreak(70_H) as RxSTATUS, MediaLB sets the GB bit in Channel n Status Configuration Register (CSCRn) to "1", detects an AsyncStart(20_H) or ControlStart(30_H) command and, if the received address agrees, changes the GB bit in Channel n Status Configuration Register (CSCRn) to "0" to output ReceiverBreak(70_H). The data received at this point is stored in the local channel buffer. The start address in which the packet data is stored is reloaded into SCA bit in the channel n current buffer configuration register (CCBCRn), and when AsyncStart (20_H) or ControlStart (30_H) is received, the reception data is stored from that address.

<Notes>

- Setting the GB bit in Channel n Status Configuration Register (CSCRn) to "1" with the local channel buffer full does not cause ReceiverBreak(70_H) to be output.
 - Transmitting ReceiverBreak(70_H) causes BCA in Channel n Current Buffer Configuration Register (CCBCRn) to be reloaded, starting to receive data from the beginning. As the FIFO buffer retains received data in spite of it sending ReceiverBreak(70_H), read successful packets from an appropriate channel in the FIFO buffer to perform software reset on that channel.
-

■ Transmitting ReceiverProtocolError(72_H)

If MediaLB receives a command that is not in the command sequence or is not allowed for the channel type after receiving AsyncStart(20_H) for async channels or ControlStart(30_H) for control channel, MediaLB sets the STS[0] bit in Channel n Status Configuration Register (CSCRn) to "1", if ping-pong buffering remains active, to output ReceiverProtocolError(72_H). The data received at this point is stored in the local channel buffer. The start address in which the packet data is stored is reloaded into SCA bit in the channel n current buffer configuration register (CCBCRn), and when AsyncStart (20_H) or ControlStart (30_H) is received, the reception data is stored from that address.

<Notes>

- Unless ping-pong buffering is active, the STS[0] bit in Channel n Status Configuration Register (CSCRn) is not to set to "1" when a protocol error is detected. Note that ReceiverProtocolError(72_H) is output. If the data that has caused a protocol error is stored in the local channel buffer with ping-pong buffering inactive, activating ping-pong buffering causes the STS[0] bit in Channel n Status Configuration Register (CSCRn) to be set to "1".
- Detecting a protocol error causes BCA in Channel n Current Buffer Configuration Register (CCBCRn) to be reloaded, starting to receive data from the beginning. As the FIFO buffer retains received data in spite of its transmitting ReceiverProtocolError(72_H), perform software reset on an appropriate channel in the FIFO buffer by the time the next data is received.

■ Transmitting ReceiverBusy(10_H)

When the local channel buffer becomes full, MediaLB outputs ReceiverBusy(10_H) to the channel address it has received. At this time, the received data is discarded.

The local channel buffer becomes full only when the ping-pong buffering remain inactive.

(4) Receiving through a Synch Channel

The setting method for sync channels is the same as that for async channels, except that, for the sync channels, the frame received first does not contain the number of bytes in the frame, which means the first frame contains only received data.

MediaLB does not output ReceiverBusy(10_H), ReceiverBreak(70_H) or ReceiverProtocolError(72_H) as RxSTATUS, but when it receives an unsupported command, (resulting in a protocol error), the STS[0] bit in Channel n Status Configuration Register (CSCRn) is set to "1". At this time, data continues to be received without the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) being reloaded with the start address.

<Note>

- As long as the ping-pong buffering remains inactive, detection of a protocol error does not cause the STS[0] bit in Channel n Status Configuration Register (CSCRn) to be set to "1". If the local channel buffer contains the data that has caused a protocol error with the ping-pong buffering remaining inactive, the STS[0] bit in Channel n Status Configuration Register (CSCRn) is set to "1" when activating the ping-pong buffering.

■ Circular Buffering

In the circular buffering, a buffer from a start address to an end address is used in a loop indefinitely by setting the start and end addresses in Next Buffer Configuration Register until the RDY bit in Channel n Status Configuration Register (CSCRn) is set to "0". This type of buffering should be used only for synch channels and is useful to exchange sync data with I²S via the FIFO buffer.

Make the following settings before using the circular buffering

- Setting of the device configuration register (DCCR)
- Setting of the channel n entry configuration register (CECRn)
- Setting of the local channel n buffer configuration register (LCBCRn)
- Setting of the FIFO buffer channel n address range register (BUFARn)
- Setting of the access select register (ASLR)
- Setting of the FIFO buffer direction control register (BUFDCR)
- Setting of the FIFO buffer n control register (BUFnCR)
- Setting I²S Common Control Register (I2SCCR)
- Setting I²S Rate Setting Register (I2SRSR)
- Setting I²S Shift Control Register (I2SSCRn)

<Note>

- Use the circular buffering only when I²S is used with synch channels.
-

(1) For Transmitting

To transmit a MediaLB frame, it is necessary to match the reading/writing directions for MediaLB and for I²S with the sending/writing direction for the FIFO buffer by setting the TR bit in Channel n Entry Configuration Register (CECRn) to "1", the ASn[1:0] bit in FIFO Access Select Register (ASLR) to "01", BRDn in FIFO Buffer Direction Control Register (BUFDCR) to "1" and the TR bit in I²S Shift Control Register (I2SSCRn) to "0". If made a mistake, the following bit(s) is (are) set to "1".

- If the directions of the MediaLB and FIFO buffer are incorrect:
The host error bit in the channel n status configuration register (CSCRn) of the MediaLB and the host error bit in the FIFO buffer error register (BUFER) are set to "1".
- If the directions of the I²S and FIFO buffer are incorrect:
The error bit in the FIFO buffer error register (BUFER) is set to "1".

Perform the following steps to set up MediaLB so that it performs circular buffering.

- (a) Set the address corresponding to the channel address that FIFO buffer uses in the channel n next buffer configuration register (CNBCRn).
 - Set the same value as ST bit in the FIFO buffer channel n address range register (BUFARn) to BSA bit in the channel n next buffer configuration register (CNBCRn), and set the address to be circulated in BEA bit. In the setting, make sure to satisfy the condition that CNBCRn:BEA ≤ BUFARn:EA.
- (b) Write "1" to RDY bit in the channel n status configuration register (CSCRn).
- (c) Write "1" to I2SRQE bit in the FIFO buffer direction control register (BUFDCR), and write "1" to START bit and RCTN bit in the I²S common control register (I2SCCR).

When the above settings are made, if it is a circular buffering, the value of the channel n next buffer configuration register (CNBCRn) is copied to the channel n current buffer configuration register (CCBCRn). At that point, STS[3] (current buffer start) bit in the channel n status configuration register (CSCRn) is set to "1" to start circular buffering and RDY bit in the channel n status configuration register (CSCRn) is kept at "1".

Once the circular buffering starts, the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is

output as an address of the HBI bus, MediaLB selects a FIFO buffer channel which falls within the range of the value of FIFO Buffer Channel n Address Range Register (BUFARn) in HB Decoder, and read transmitted data from the selected FIFO buffer channel in a fixed length of 32 bits. Once read is complete, MediaLB stores the read data in the local channel buffer, increasing the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) by +4. When the value of the BCA bit reaches that of the BFA bit, the value of Channel n Next Buffer Configuration Register (CNBCRn) is copied to Channel n Current Buffer Configuration Register (CCBCRn). The data received by I²S is written to the FIFO buffer, and then output as transmitted data to MediaLB.

To stop the circular buffering, stop receiving data from I²S and write "1" to the RDY bit in Channel n Status Configuration Register (CSCRn) for MediaLB, which changed the same to "0". When the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) accesses the address indicated by the BFA bit, STS[2] in Channel n Status Configuration Register (CSCRn) is "1", which indicates the circular buffering is no longer active. Then, software reset is performed on the appropriate channels in the FIFO buffer.

<Notes>

- Make sure to set the MediaLB and I²S so as to match the transfer rates of one frame of the MediaLB and one frame of the I²S.
 - If MediaLB transmits a plurality of sub frames that consist of one frame in succession via the same channel, (00_H) data is inserted as data transmitted from MediaLB, as I²S is too busy to receive such data. So, assign one MediaLB frame to one channel address.
-

(2) For Receiving

To receive a MediaLB frame, it is necessary to match the reading/writing directions for MediaLB and for I²S with the sending/writing direction for the FIFO buffer by setting the TR bit in Channel n Entry Configuration Register (CECRn) to "0", the ASn[1:0] bit in FIFO Access Select Register (ASLR) to "01", BRDn in FIFO Buffer Direction Control Register (BUFDCR) to "0" and the TR bit in I²S Shift Control Register (I2SSCRn) to "1". If made a mistake, the following bit(s) is (are) set to "1".

- If the directions of the MediaLB and FIFO buffer are incorrect:
The host error bit in the channel n status configuration register (CSCRn) of the MediaLB and the host error bit in the FIFO buffer error register (BUFER) are set to "1".
- If the directions of the I²S and FIFO buffer are incorrect:
The error bit in the FIFO buffer error register (BUFER) is set to "1".

Perform the following steps to set up MediaLB so that it performs circular buffering.

- Set the address corresponding to the channel address that FIFO buffer uses in the channel n next buffer configuration register (CNBCRn).
 - Set the same value as ST bit in the FIFO buffer channel n address range register (BUFARn) to BSA bit in the channel n next buffer configuration register (CNBCRn), and set the address to be circulated in BEA bit. In the setting, make sure to satisfy the condition that CNBCRn:BEA ≤ BUFARn:EA.
- Write "1" to RDY bit in the channel n status configuration register (CSCRn).
- Write "1" to I2SRQE bit in the FIFO buffer direction control register (BUFDCR), and write "1" to START bit and RCTN bit in the I²S common control register (I2SCCR).

When the above settings are made, if it is a circular buffering, the value of the channel n next buffer configuration register (CNBCRn) is copied to the channel n current buffer configuration register (CCBCRn). At that point, STS[3] (current buffer start) bit in the channel n status configuration register (CSCRn) is set to "1" to start circular buffering and RDY bit in the channel n status configuration register (CSCRn) is kept at "1".

When received channel address agrees after circular buffering starts, the received data is stored in the local channel buffer temporarily. Then, the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is

stored as an HBI bus address, and the received data is output as data on the HBI bus. A FIFO buffer channel which falls within the range of the value of FIFO Buffer Channel n Address Range Register (BUFARn) in HB Decoder is selected, the data is written in the selected FIFO buffer channel in a fixed length of 32 bits, and the value of the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is increased by 4. When the value of the BCA bit reaches that of the BFA bit, the value of Channel n Next Buffer Configuration Register (CNBCRn) is copied to Channel n Current Buffer Configuration Register (CCBCRn). The data received by MediaLB is written in the FIFO buffer and then output to I²S as transmitted data.

To stop circular buffering, write "1" to the RDY bit in Channel n Status Configuration Register (CSCRn) for MediaLB, which changes the same to "0". When the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) accesses the address indicated by the BFA bit, STS[2] in Channel n Status Configuration Register (CSCRn) is "1", which indicates that the circular buffering is no longer active. Then, terminate I²S and perform software reset on appropriate channels in the FIFO buffer.

<Notes>

- Make sure to set the MediaLB and I²S so as to match the transfer rates of one frame of the MediaLB and one frame of the I²S.
 - If MediaLB receives a plurality of sub frames that consist of one frame in succession via the same channel, I²S is too busy to transmit such data, causing an overrun error in the FIFO buffer. Assign one MediaLB frame to one channel address.
-

60.4.3 Access Selection

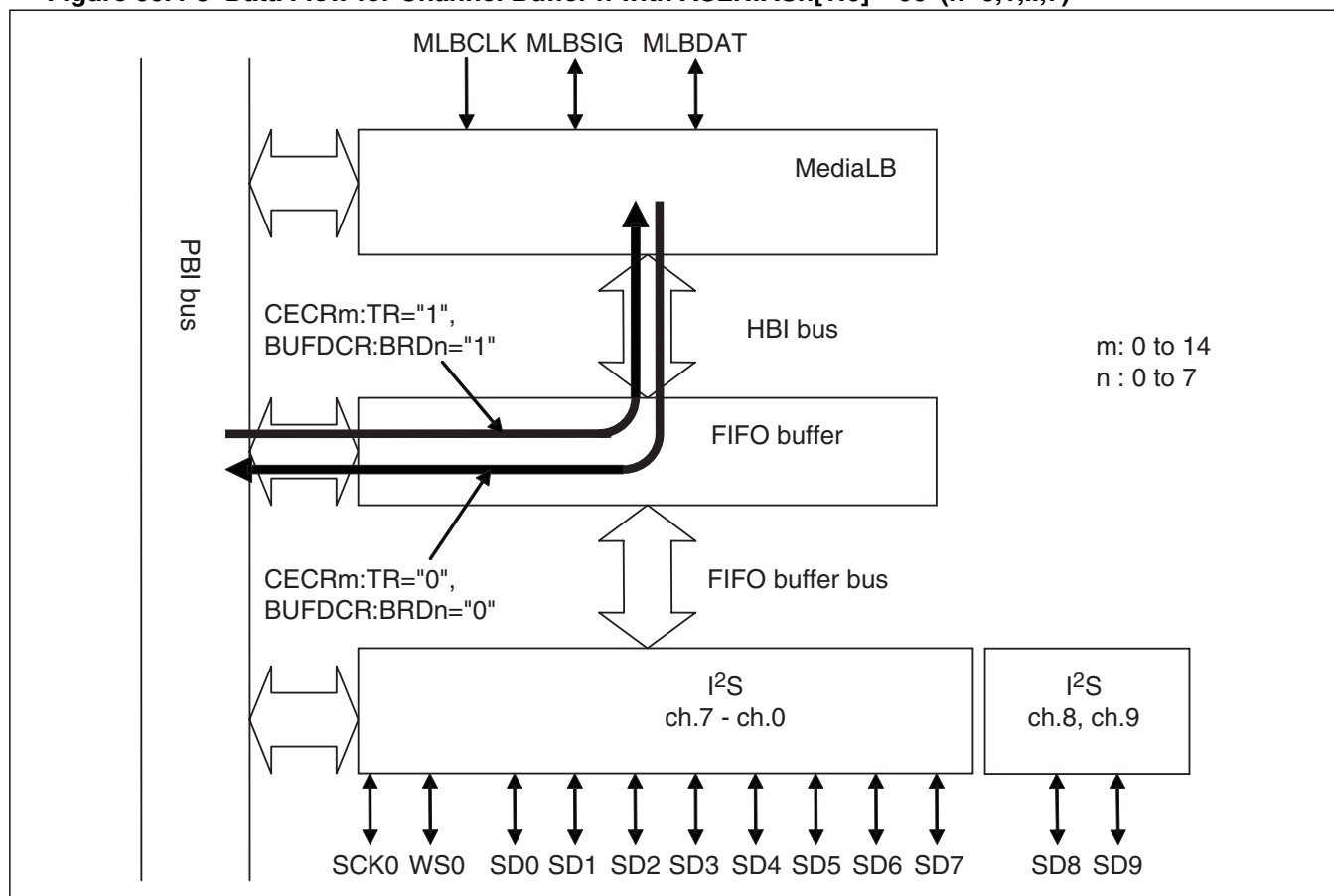
As FIFO buffer's channel buffers 7- 0 are accessible from three directions (MediaLB, IS2, and software), Access Select Register can be used to determine which, MediaLB, IS2, or software, can access these buffers.

■ Transmission/Reception data flow

(1) Channel Buffers 7 – 0

Channel buffers 7 – 0 have the following data flow, depending on the setting of Access Select Register (ASLR).

- Data flow for channel buffer n with ASLR:ASn[1: 0]="00" (n=0,1,...,7)

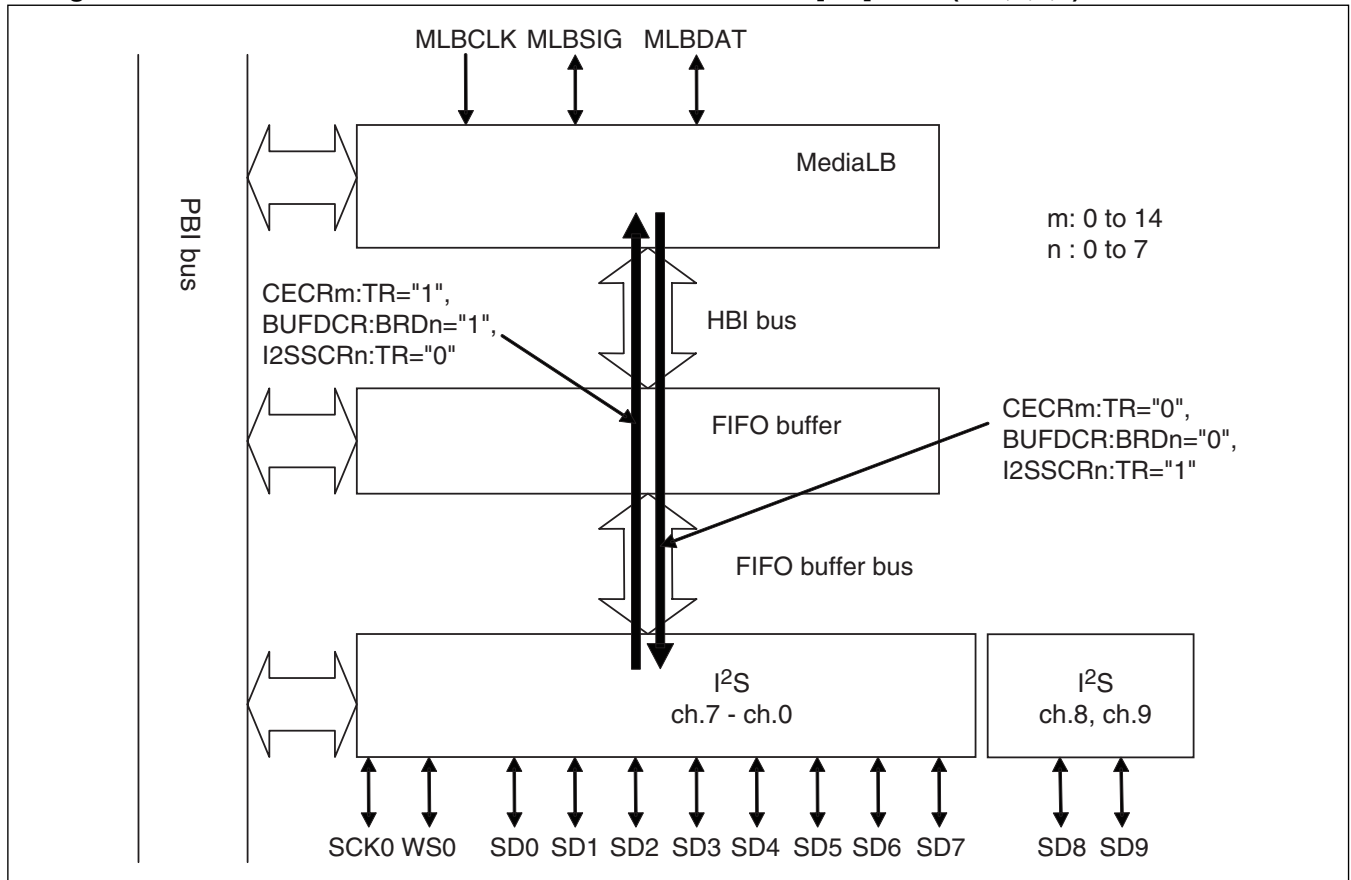
Figure 60.4-3 Data Flow for Channel Buffer n with ASLR:ASn[1:0]="00"(n=0,1,...,7)

<Notes>

- Match the transmitting/writing direction for the FIFO buffer with the reading/writing direction for MediaLB.
- Ch.8 and ch.9 in the I²S can be accessed only form PBI bus regardless of ASLR:ASn[1:0] setting.

MB91460 Series

- Data flow for channel buffer n with ASLR.ASn[1:0]="01" (n=0,1,...,7)

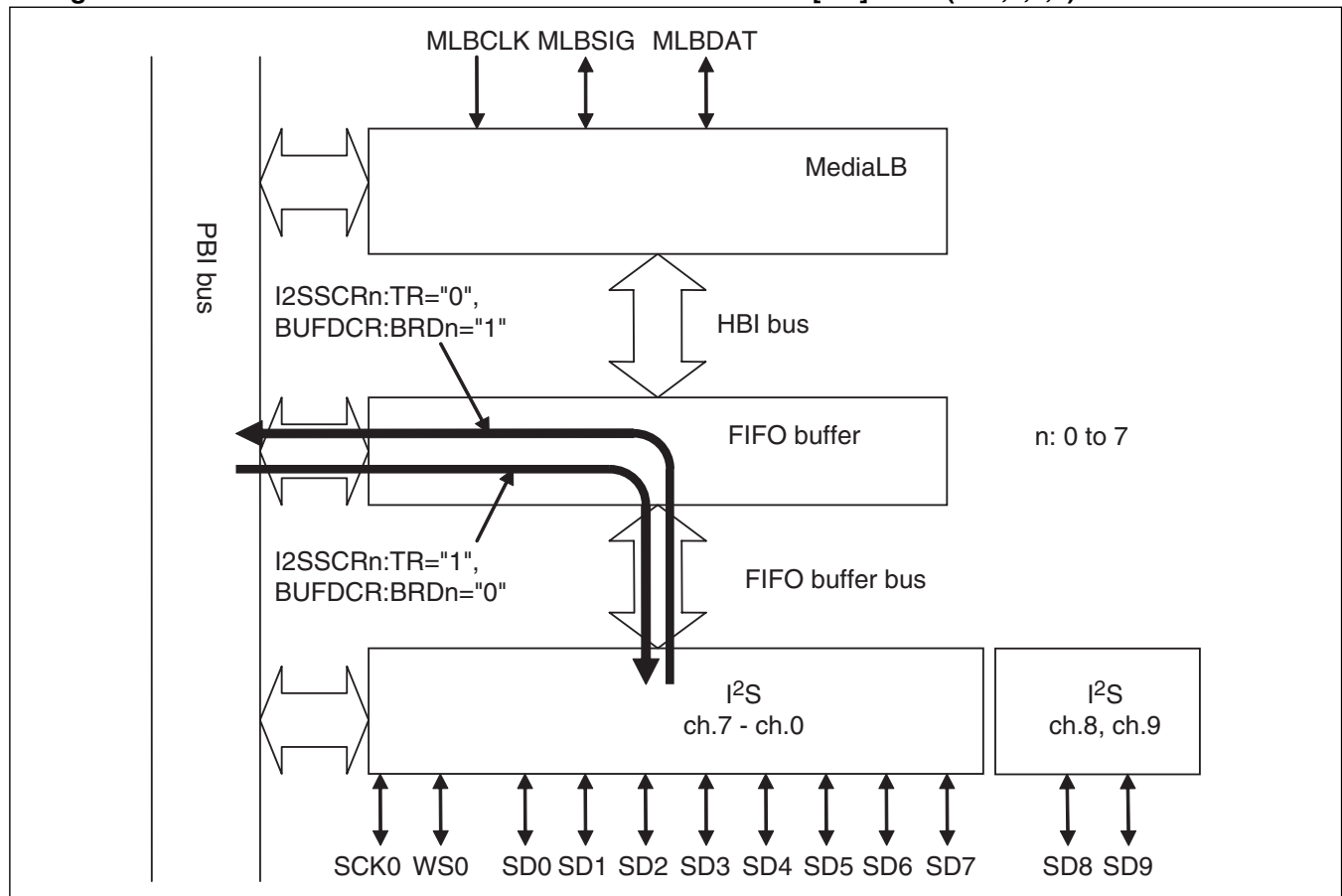
Figure 60.4-4 Data Flow for Channel Buffer n with ASLR.ASn[1:0]="01" (n=0,1,...,7)

<Notes>

- Match the reading/writing directions for MediaLB and for I²S with the transmitting/writing direction for the FIFO buffer.
- Ch.8 and ch.9 in the I²S can be accessed only form PBI bus regardless of ASLR.ASn[1:0] setting.

- Data flow for channel buffer n with ASLR.ASn[1:0]="10" (n=0,1,...,7)

Figure 60.4-5 Data Flow for Channel Buffer n with ASLR.ASn[1:0]="10" (n=0,1,...,7)



<Notes>

- Match the transmitting/writing direction for the FIFO buffer with the reading/writing direction for I²S.
- Ch.8 and ch.9 in the I²S can be accessed only from PBI bus regardless of ASLR.ASn[1:0] setting.

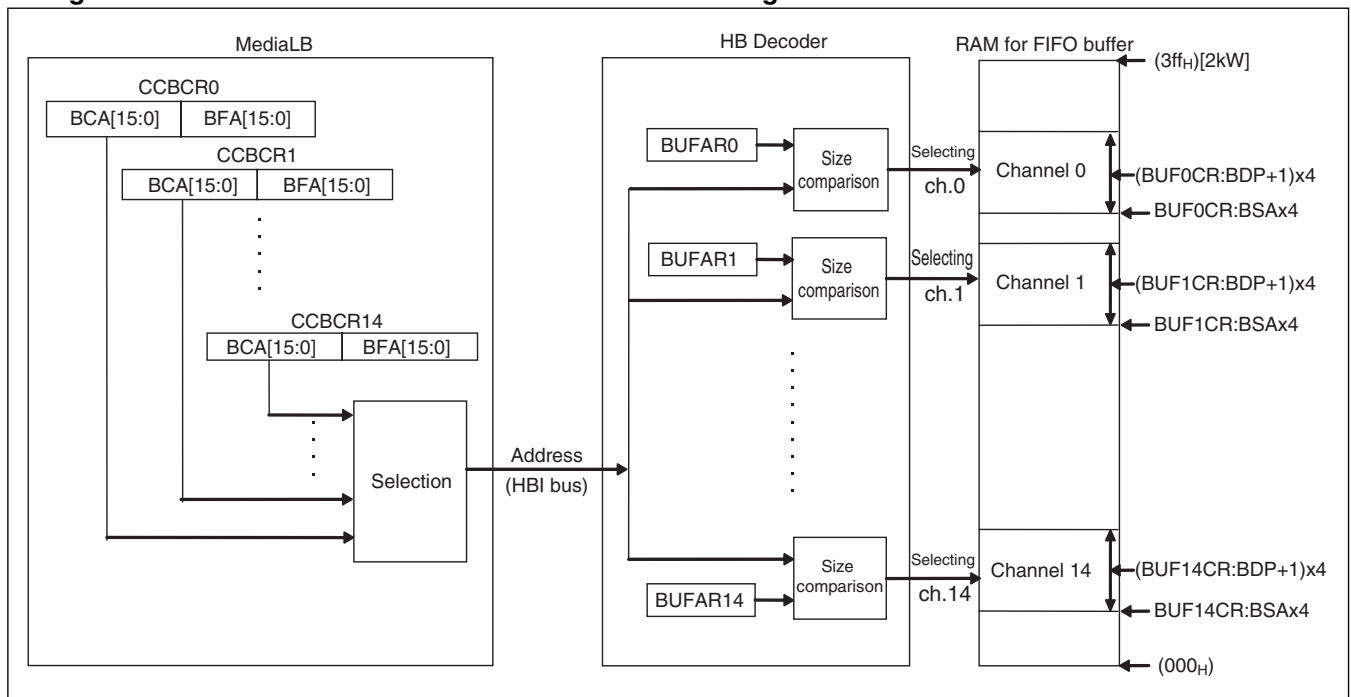
60.4.4 Link between MediaLB and FIFO Buffer Registers

The FIFO buffer comprises RAM of 2048 words × 32 bits or 1024 words × 32 bits, and is connected to MediaLB via the HBI bus. For this reason, MediaLB and the FIFO buffer have registers that are set so that they have connection.

The FIFO buffer can support up to 15 channels, of which Channels 0 - 7 can exchange data with I²S. These 15 channels correspond to MediaLB's 15 channels.

■ FIFO Buffer and MediaLB Settings

To use the FIFO buffer, FIFO Buffer n Control Register (BUF_nCR) for the FIFO buffer, FIFO Buffer Channel n Address Range Register (BUF_{AR}n) in HB Decoder and Channel n Current Buffer Configuration Register (CCBC_Rn) for MediaLB need to be linked with each other. [Figure 60.4-6](#) shows links between the registers.

MB91460 Series**Figure 60.4-6 Link between MediaLB and FIFO Buffer Registers**

Once the DMA mode is activated, the BCA bit in Channel n Current Buffer Configuration Register (CCBCRn) is output as the address of the HBI bus. Whether the address falls within the range indicated by the SA and EA bits in FIFO Buffer Channel n Address Range Register (BUFARn) is determined. If it is within the range, the channel buffer in the FIFO buffer is selected. The selected channel buffer has an internal pointer to store data in an area that starts at the address indicated by the BSA bit in FIFO Buffer n Control Register (BUFnCR) whose range is indicated by the BDP bit.

Set the registers so that the following conditions are met:

- Setting requirements for FIFO Buffer Channel n Address Range Register (BUFARn)
 - [1] BUFARn:ST[15:0] = CCBCRn:BCA[15:0]
CCBCRn:BCA[15:0] indicates an address when the STS[3] bit in Channel n Status Configuration Register (CSCRn) is set to "1"
 - [2] BUFARn:EA[15:0] ≥ CCBCRn:BFA[15:0]
Set BUFARn:EA to a value larger than CCBCRn:BFA, taking the number of bytes received by MediaLB during ping-pong buffering into account, as it may exceed CCBCRn:BFA.
- Setting requirements for FIFO Buffer n Control Register (BUFnCR)
 - [1] BUFnCR:BSA ≥ BUFnCR:BDP + 1 × 4
m indicates the number of a channel for which a RAM is assigned before n.
 - [2] BUFnCR:BDP = (BUFARn:EA-BUFARn:ST)/16
Round down the digits below the decimal point.
 - [3] BUFnCR:BSA + (BUFnCR:BDP + 1) × 4 ≤ RAM word count
n indicates the number of a channel for the last RAM area (having the largest RAM address).

60.4.5 Dual Transfer Mode

Channels 8 and 9 of MediaLB, the FIFO buffer, and I²S support the dual transfer mode in which the same data is transmitted twice.

The dual transfer mode is a mode in which the same data is transmitted twice when writing the transmitted data. When the DBL bit in I²S Common Control Register (I2SCCRn) is set to "1", the data written to MediaLB Channel n Next Buffer Configuration Register (CNBCRn) for Channels 8 and 9, FIFO Buffer Channel n Data Register (BUFnDTR) for Channels 8 and 9, and I²S Left and Right Registers (LTDTn, RTDTn) for Channels 8 and 9 is transmitted twice.

When the FIFO buffer and MediaLB write data to their registers, the data is written twice, increasing the number of valid data pieces by 2. On the other hand, in I²S, which has an internal flag, writing data in the dual transfer mode causes the flag to be reset to "1", thus preventing I²S from writing twice, which results in the number of valid data pieces being increased by 1.

<Note>

To put MediaLB in the DMA mode to use the dual transfer mode, set the DBL bit in I²S Common Control Register (I2SCCR) to "0" when writing to Channel n Next Buffer Configuration Register (CNBCRn).

60.4.6 Local channel buffer

The local channel buffer comprises RAM of 1024 words × 32 bits or 2048 words × 32 bits. The local channel buffer is in MediaLB and used to store transmitted and received data temporarily.

A local channel buffer is used to assign a RAM area to each logical channel. Each logical channel uses that area to transmit and receive data.

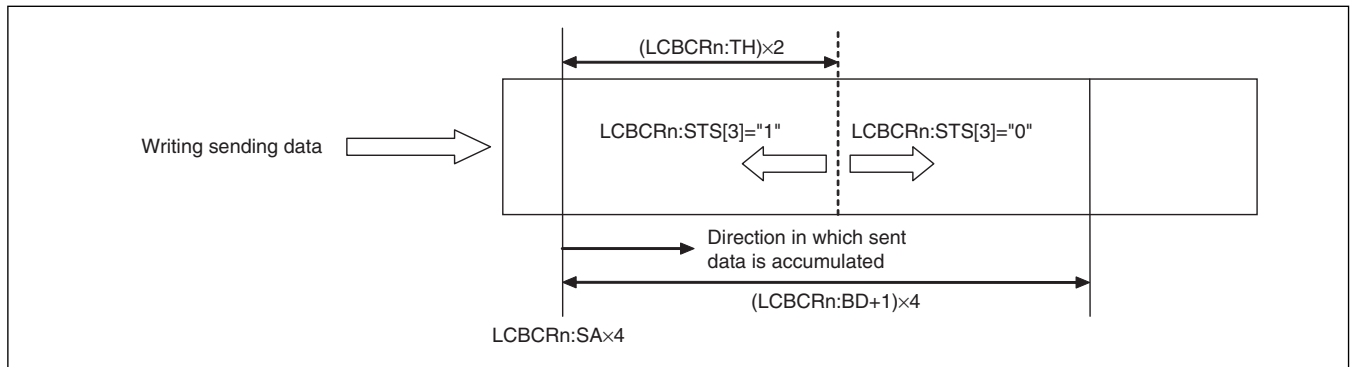
To assign a RAM area, Local Channel n Buffer Configuration Register (LCBCRn) is used to set the start address (SA bit) and a length (BD bit) of the RAM area. For the setting method, refer to "60.3.11 Local Channel n Buffer Configuration Register (LCBCRn)".

MB91460 Series

Furthermore, in the IO mode, an interrupt can be triggered to each logical channel by setting a threshold value (TH bit) using Local Channel n Buffer Configuration Register (LCBCRn).

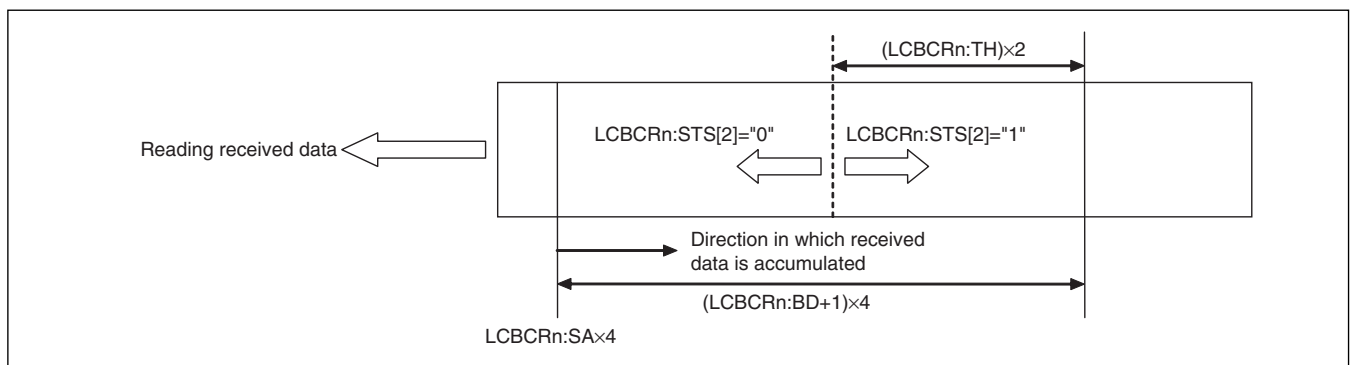
■ When MediaLB Transmits Data (CECRn:TR="1")

As shown in the following diagram, when the number of valid quadlets (4 byte data) in the local channel buffer becomes less than the threshold value for transmitting data, a send service request is made.



■ When MediaLB Receives Data (CECRn:TR="0")

As shown in the following diagram, when the number of valid quadlets (4 byte data) in the local channel buffer exceeds the threshold value for receiving data, a receive service request is made.



60.4.7 Setting Channel Buffers for FIFO Buffer

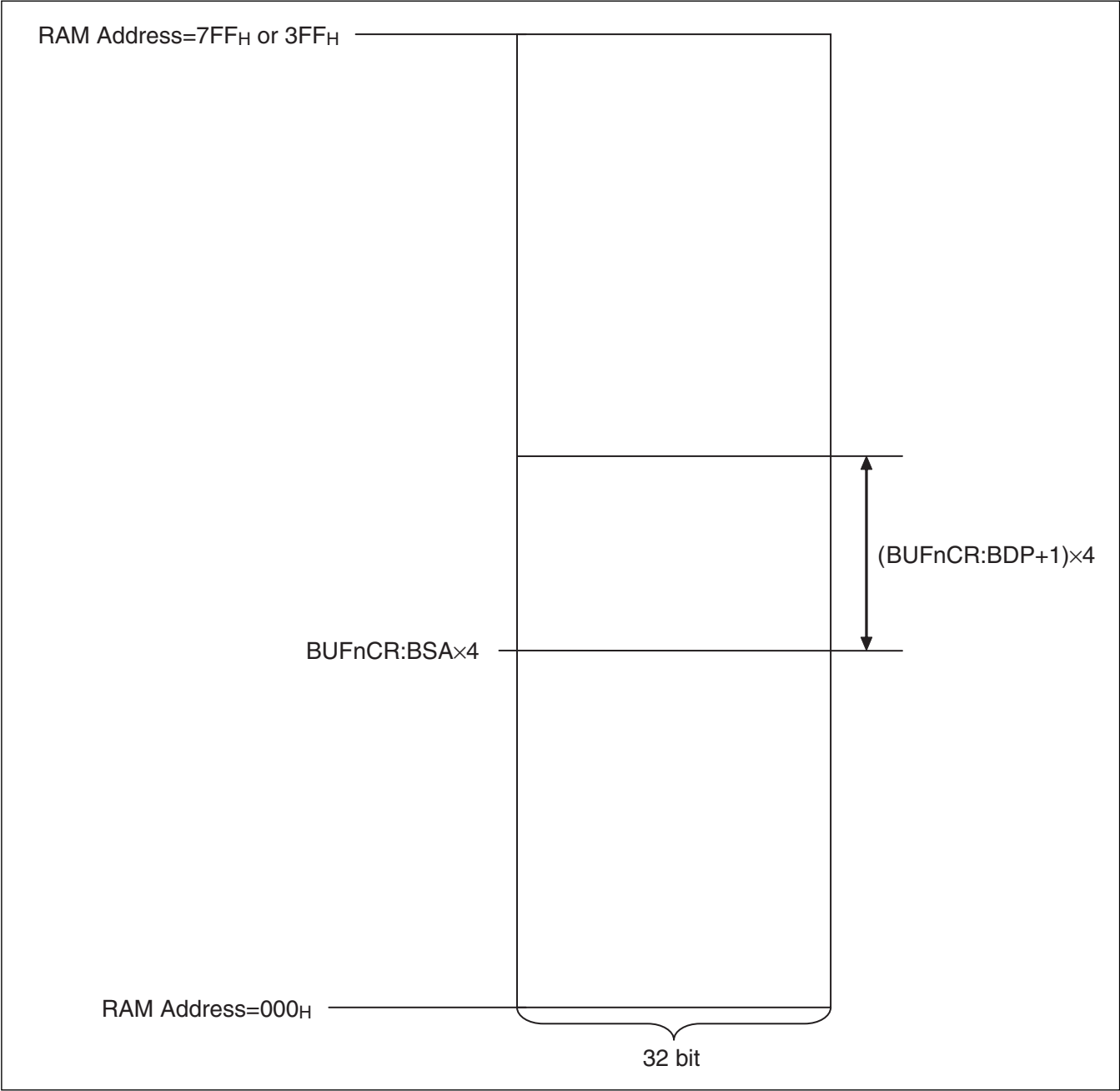
The FIFO buffer comprises RAM of 1024 words × 32 bits or 2048 words × 32 bits. A channel buffer is divided into areas each of which corresponds to a logical channel. The channel buffer can be defined by setting the start address and its length. Channel buffers 0 – 7 can be accessed from I²S.

■ Setting a Channel Buffer

The channel buffer can be defined using the BSA and BDP bits in Buffer n Control Register (BUFnCR).

The following diagram shows the relationship between RAM and a channel buffer.

Figure 60.4-7 Relationship between RAM and Channel Buffer



An interrupt can be generated to each channel buffer by setting a threshold (BTH bit) using FIFO Buffer n Control Register (BUF_nCR)

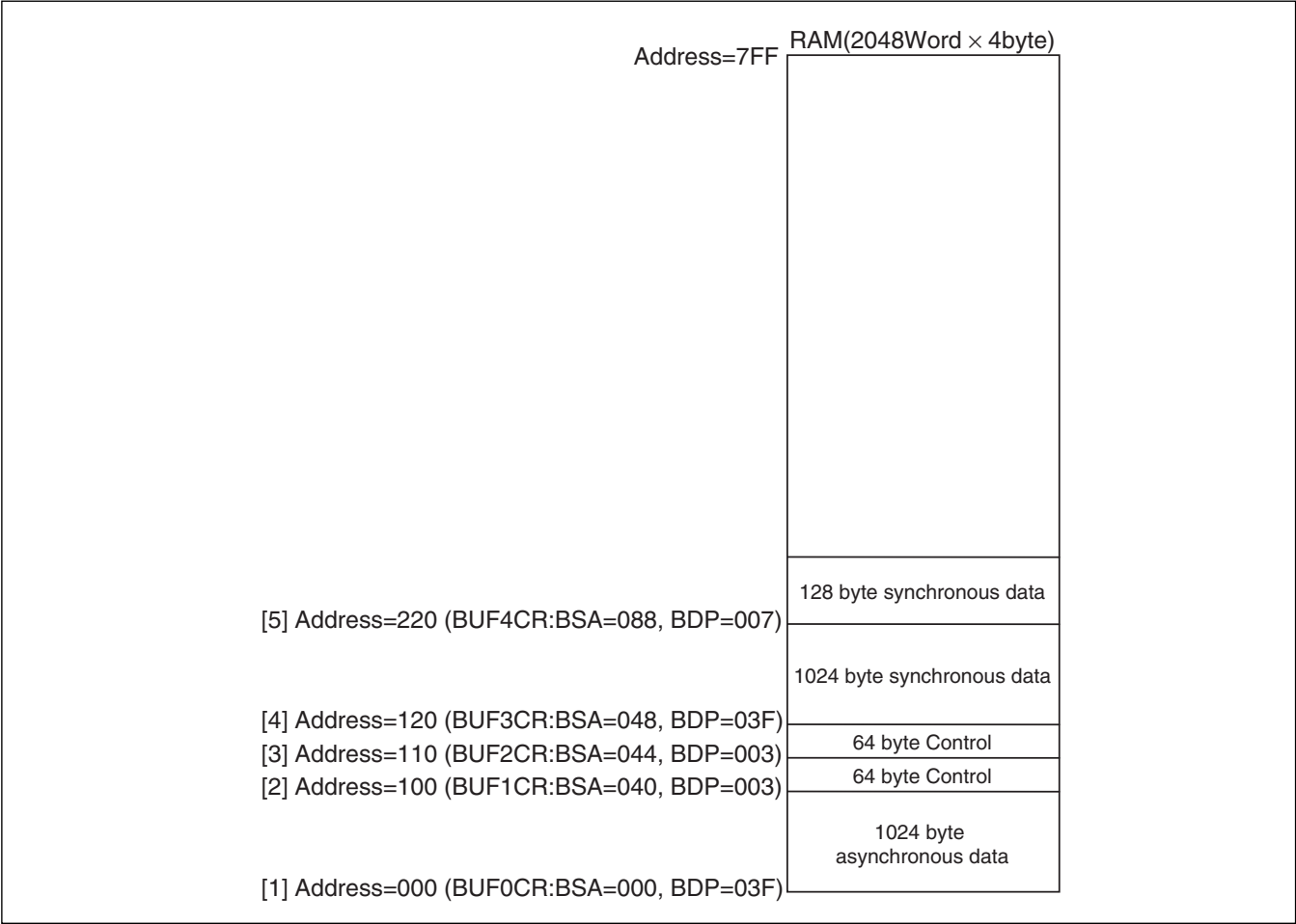
Example of Setting Channel Buffer Start Address:

To assign a 1024 byte asynchronous data area for channel buffer 0, two 64 byte control data areas with one for channel buffer 1 and the other for channel buffer 2, a 1024 byte synchronous data area for channel buffer 3, and a 128 byte synchronous data area for channel buffer 4, starting from a lower RAM address, set Buffer n Control Register (BUF_nCR) as follows:

- Calculation method:

- [1] 1024 byte asynchronous data
 RAM Start address= 000
 BUF0CR:BSA= 000
 BUF0CR:BDP= $1024\text{byte}/4\text{byte}/4 - 1 = 63(0x03f)$
- [2] 64 byte control data
 RAM Start address= Following [1] data area = $1024\text{byte}/4\text{byte} = 256(0x100)$
 BUF1CR:BSA= $256/4 = 64(0x040)$
 BUF1CR:BDP= $64\text{byte}/4\text{byte}/4 - 1 = 3(0x003)$
- [3] 64 byte control data
 RAM Start address= Following [1] and [2] data areas = $256+64\text{byte}/4\text{byte} = 272(0x110)$
 BUF2CR:BSA= $272/4 = 68(0x044)$
 BUF2CR:BDP= $64\text{byte}/4\text{byte}/4 - 1 = 3(0x003)$
- [4] 1024 byte synchronous data
 RAM Start address: Following [1], [2] and [3] data areas = $272+64\text{byte}/4\text{byte} = 288(0x120)$
 BUF3CR:BSA: $288/4 = 72(0x048)$
 BUF3CR:BDP: $1024\text{byte}/4\text{byte}/4 - 1 = 63(0x03f)$
- [5] 128 byte synchronous data
 RAM Start address: Following [1], [2], [3] and [4] data areas = $288 + 1024\text{byte}/4\text{byte} = 544(0x220)$
 BUF4CR:BSA: $544/4 = 136(0x088)$
 BUF4CR:BDP: $128\text{byte}/4\text{byte}/4 - 1 = 7(0x007)$

Figure 60.4-8 Setting Example of RAM and Buffer n Control Register (BUFnCR)



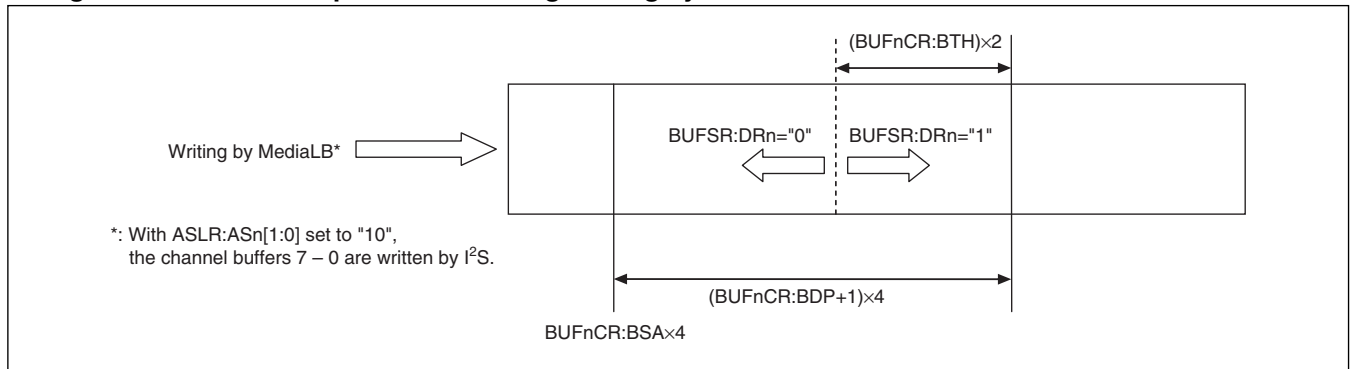
MB91460 Series

■ Writing by MediaLB

When `BUFIER:BIREn="1"`, an interrupt is triggered during writing by MediaLB when the following condition is met.

- $(\text{BUFnCR:BDP} \times 4 - \text{BUFnCR:BTH} \times 2 + 4) \leq \text{BUFCTm:BCTn}$

Figure 60.4-9 Data Request Made during Writing by MediaLB

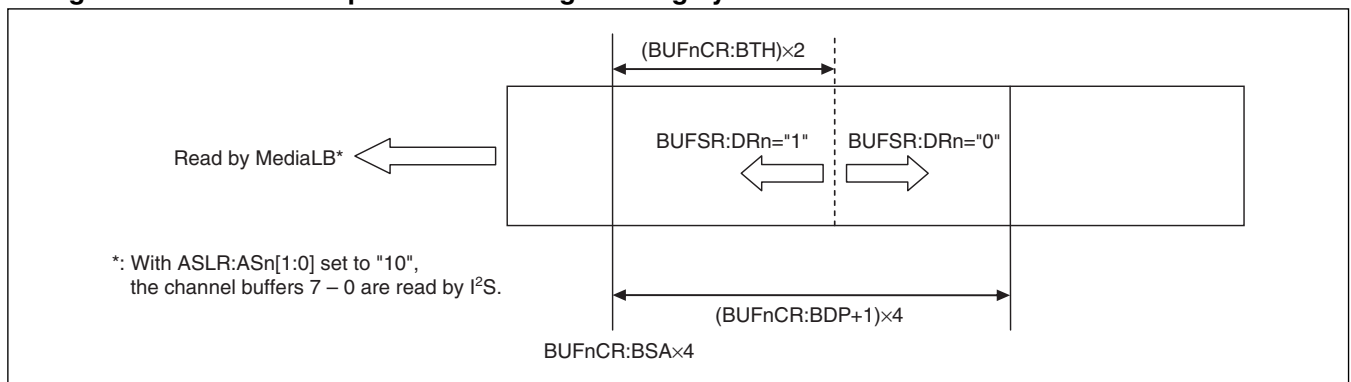


■ Reading by MediaLB

When `BUFIER:BIREn="1"`, an interrupt is triggered during reading by MediaLB when the following condition is met.

- $(\text{BUFnCR:BTH}) \times 2 \geq \text{BUFCTm:BCTn}$

Figure 60.4-10 Data Request Made during Reading by MediaLB



60.4.8 Loop Back Mode

MediaLB supports the loopback mode in which data received via Logical Channel 1 is transmitted back via the same.

To facilitate debugging of transmission paths, MediaLB supports the loopback test mode.

Setting the LBM bit in Device Control Configuration Register (DCCR) to "1" causes data to be received via Channel 0 and transmitted back via Channel 1.

To use the loopback test mode, use the following steps to set the mode:

- [1] Set the logical channel addresses for Channels 0 and 1. (Same address cannot be used for different channels.)
- [2] Set Channel 0 to any of the following appropriate channel types: Sync, Async, and Control data. Enable "Receipt" and set "Enable channel".

[3] Set Channel 1 to the same channel type as that of Channel 0 and set "Enable channel".

[4] Set the loopback mode bit (DCCR:LBM="1").

<Notes>

- During the loopback mode, a protocol error or a break is prohibited for both Channels 0 and 1.
- The Next Buffer Ready bits for Channels 0 and 1 remains cleared. (CSCR0:RDY=CSCR1:RDY="0")

60.5. Interrupts by MediaLB

This section describes interrupts that MediaLB has. MediaLB has three types of interrupt: Channel interrupts indicating the state of each channel buffer (Cint), System interrupts indicating the state of the MediaLB system (Sint), and FIFO buffer interrupts.

This section describes MediaLB channel interrupts (Cint), system interrupts (Sint) and FIFO buffer interrupts.

60.5.1 Channel Interrupts (Cint)

A channel interrupt is triggered by the status or error information retained by Channel n Status Configuration Register (CSCRn). The Mask bit in Channel n Entry Configuration Register (CECRn) controls whether an interrupt is triggered or not. To know in which channel an interrupt has been triggered, Channel Interrupt Configuration Register (CICR) needs to be read.

The following paragraph describes an interrupt flag for channel interrupts and interrupt cause factors. n indicates a channel number (n=14 – 0).

The interrupt flag is changed to "1" when an interrupt cause is detected. When an interrupt mask is set to "1", the interrupt is masked and not triggered by the cause factor.

There are bits that change the meaning of the interrupt flag, depending on the operation mode (IO/DMA mode).

■ Common Across IO and DMA Modes:

Interrupt Flag	Register	Interrupt Source	Valid Channel	Interrupt Mask	Interrupt Flag Clear
STS[0]	CSCRn	Current Buffer Protocol Error	All channels (async/sync/control) with reception setting, and async and control channels with transmission setting.	CECRn:MASK[0]	Writing "1" to STS[0]
STS[1]		Current Buffer Break	All channels (async/sync/control) with reception setting, and async and control channels with transmission setting.	CECRn:MASK[1]	Writing "1" to STS[1]
STS[4]		Buffer Error	Sync channels (both transmission/reception)	CECRn:MASK[4]	Writing "1" to STS[4]
STS[6]		Frame Sync Lost	Sync channels (both transmission/reception)	CECRn:MASK[6]	Writing "1" to STS[6]

MB91460 Series**■ IO Mode:**

Interrupt Flag	Register	Interrupt Source	Valid Channel	Interrupt Mask	Interrupt Flag Clear
STS[2]	CSCRn	Receive service requests	All channels enabled for receiving (async/sync/control)	CECRn:MASK[2]	Writing "1" to STS[2]
STS[3]		Transmit service request	All channels enabled for transmitting (async/sync/control)	CECRn:MASK[3]	Writing "1" to STS[3]
STS[8]		Received packet aborted	Async and control channels enabled for receiving	CECEn:MASK[2]	Writing "1" to STS[8]

■ DMA Mode:

Interrupt Flag	Register	Interrupt Source	Valid Channel	Interrupt Mask	Interrupt Flag Clear
STS[2]	CSCRN	Current buffer ends	All channels (async/sync/control) (both transmission/reception)	CECRn:MASK[2]	Writing "1" to STS[2]
STS[3]		Current buffer starts	All channels (async/sync/control) (both transmission/reception)	CECRn:MASK[3]	Writing "1" to STS[3]
STS[5]		Host bus error	All channels (async/sync/control) (both transmission/reception)	*	Writing "1" to STS[5]
STS[8]		Previous buffer protocol error	All channels enabled for receiving (async/sync/control) or async and control channel enabled for transmitting	CECEn:MASK[2]	Writing "1" to STS[8]
STS[9]		Previous buffer break	All channels (async/sync/control) (both transmission/reception)	CECRn:MASK[1]	Writing "1" to STS[9]
STS[10]		Previous buffer end	All channels (async/sync/control) (both transmission/reception)	CECRn:MASK[2]	Writing "1" to STS[10]
STS[11]		Previous buffer start	All channels (async/sync/control) (both transmission/reception)	CECRn:MASK[3]	Writing "1" to STS[11]

*: Cannot be masked. Setting the cause factor flag to "1" triggers an interrupt.

To know in which channel an interrupt has been triggered, Channel Interrupt Configuration Register (CICR) needs to be read. Each bit in Channel Interrupt Configuration Register (CICR) is cleared to 0 by clearing the factor that has triggered an interrupt to each channel or masking the interrupt using the mask bit.

60.5.2 System Interrupt (Sint)

A system interrupt is triggered when a system command, locking or unlocking is detected.

The following paragraph describes interrupt flags and interrupt cause factors for system interrupts. An interrupt flag is changed to "1" when an interrupt cause is detected. Each interrupt flag has a corresponding interrupt mask bit. Setting an interrupt mask to "1" enables masking.

Interrupt Source	Register	Interrupt Flag	Interrupt Mask	Interrupt Flag Clear
Detecting a reset (MlbRset(FE _H))	SSCR	SDR	SNCR:SMR	Writing "1" to SDR
Detecting a network lock (MOST_Lock(F0 _H))		SDNL	SMCR:SMNL	Writing "1" to SDNL
Detecting a network unlock (MOST_Unlock(E2 _H))		SDNU	SMCR:SMNU	Writing "1" to SDNU
Detecting a channel scan (MlbScan(E4 _H))		SDCS	SMCR:SMCS	Writing "1" to SDCS
Detecting a sub-command (MlbSubCmd(E6 _H))		SDSC	SMCR:SMML	Writing "1" to SDSC
Detecting a MediaLB lock		SDML	SMCR:SMML	Writing "1" to SDNL
Detecting a MediaLB unlock		SDMU	SMCR:SMMU	Writing "1" to SDNU

MB91460 Series

60.5.3 FIFO Buffer Interrupts

FIFO buffer interrupts support interrupts for data requests.

■ Interrupt Causes for FIFO Buffer

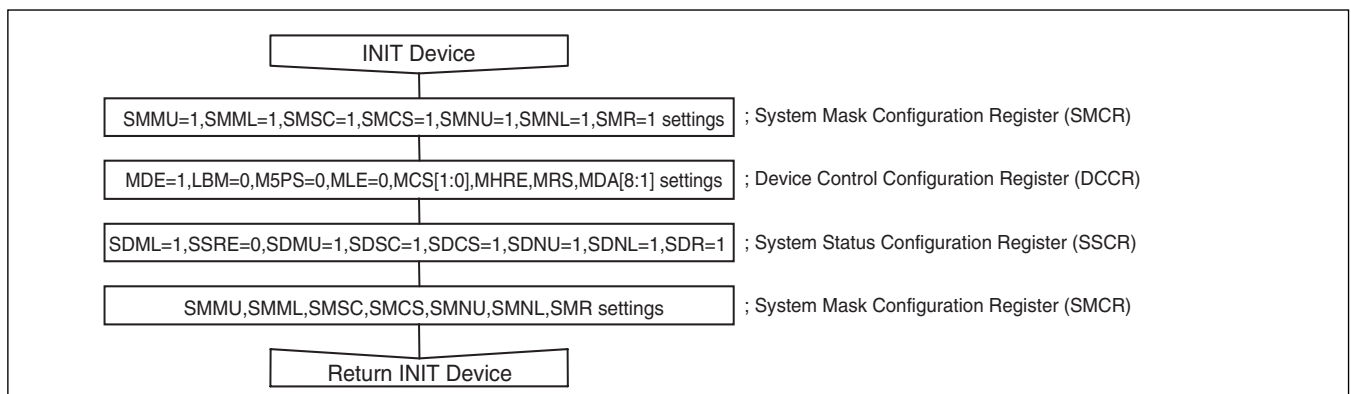
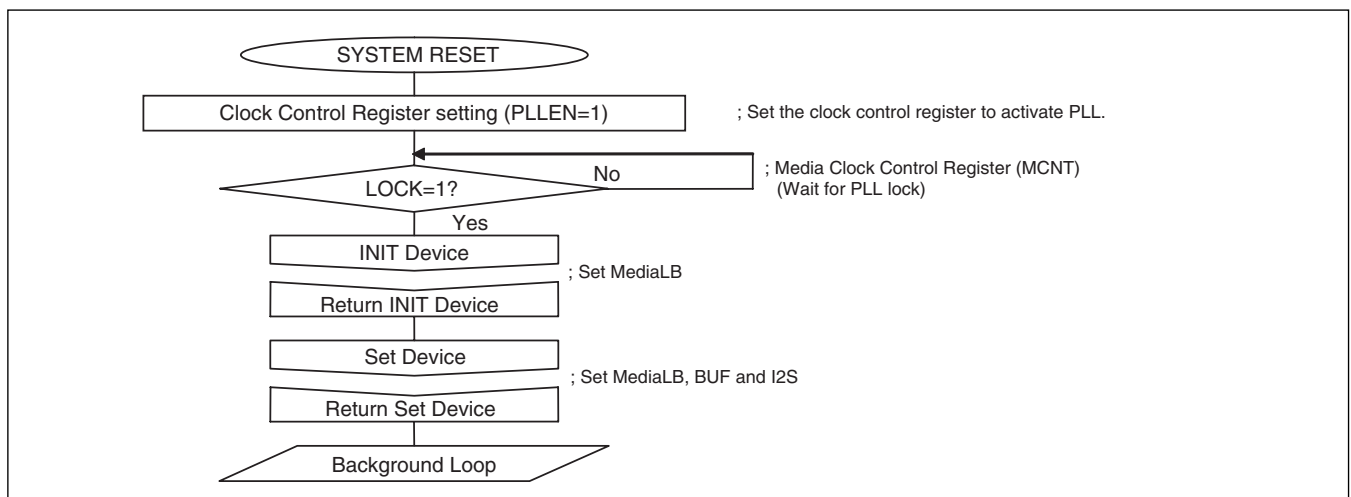
Interrupt flags and interrupt sources are shown below.

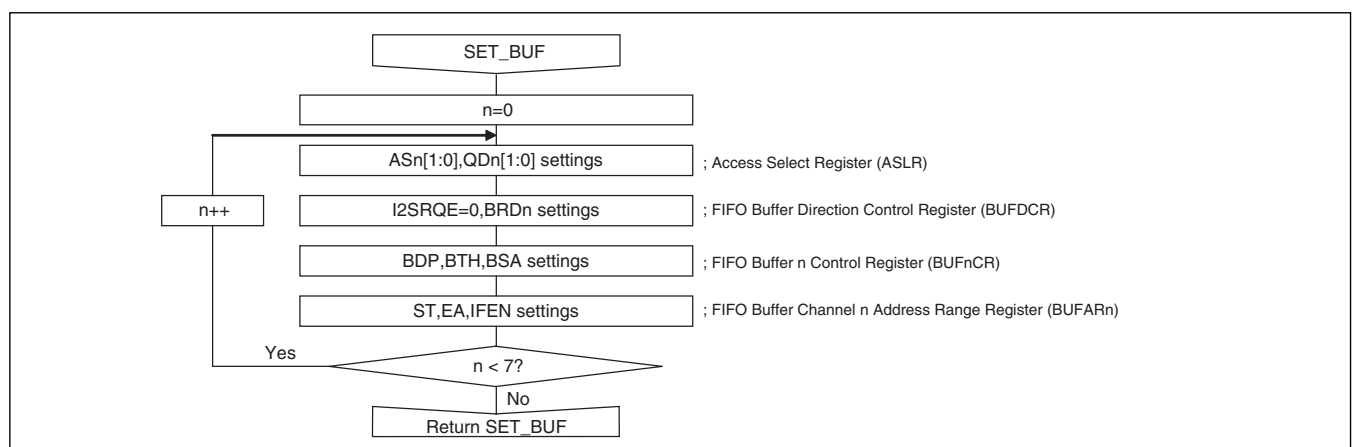
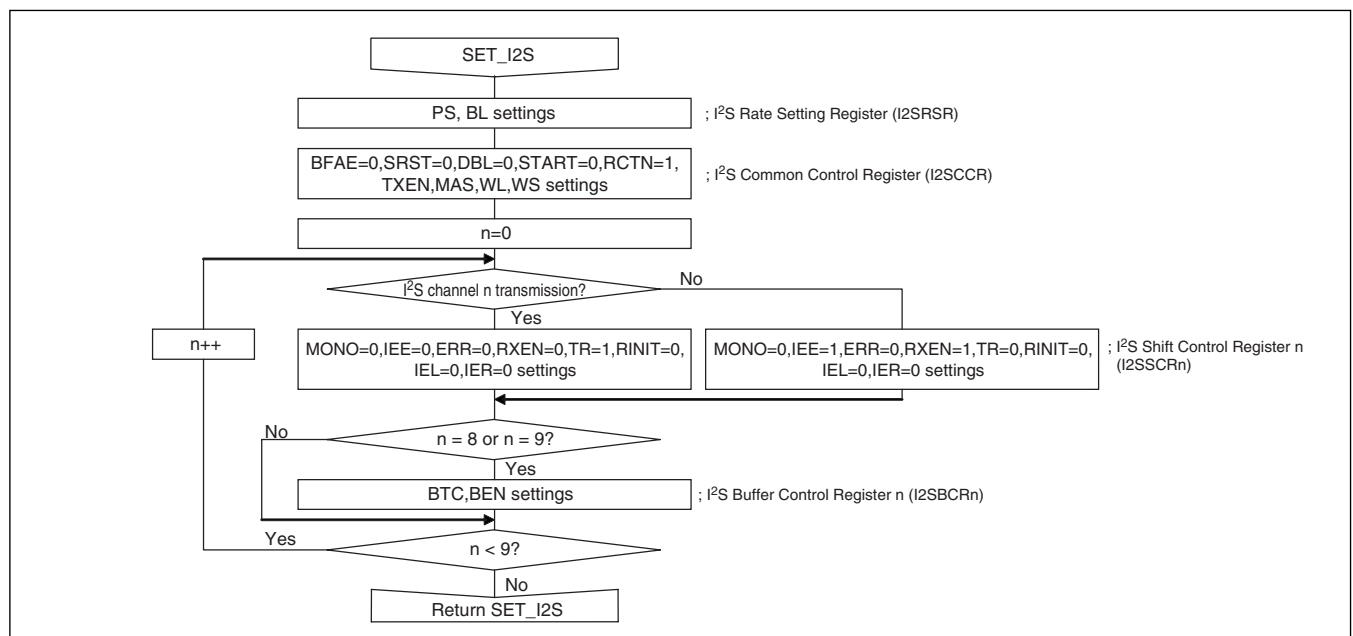
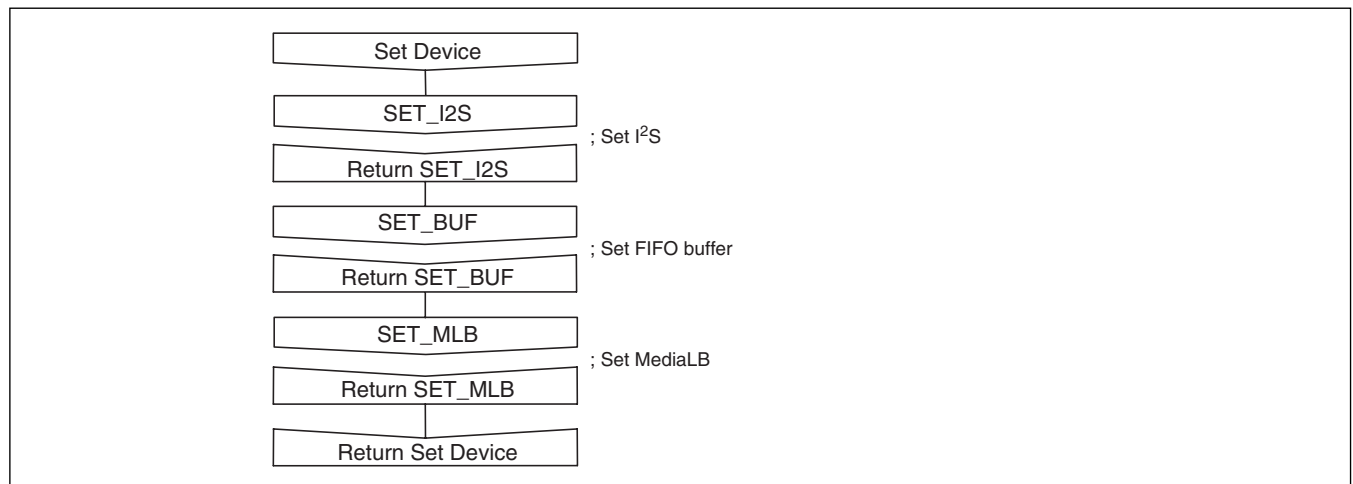
Direction	Interrupt Flag	Register	Interrupt Source	Interrupt Enable Bit	Interrupt Flag Clear
Write (BUFDCCR:BRDn="0")	DRn	BUFSR	Writing by MediaLB or I ² S	BUFIER:BIREn	Reading a channel buffer
Read (BUFDCCR:BRDn="1")	DRn	BUFSR	Reading by MediaLB or I ² S	BUFIER:BIREn	Writing to a channel buffer

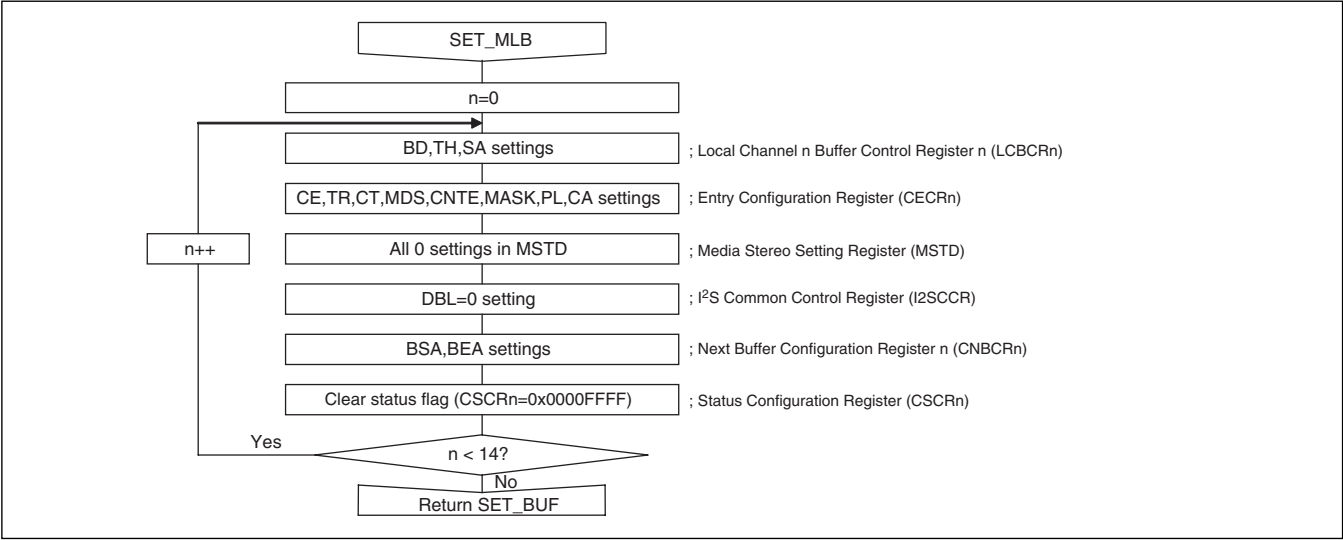
60.6. Examples of operating procedures

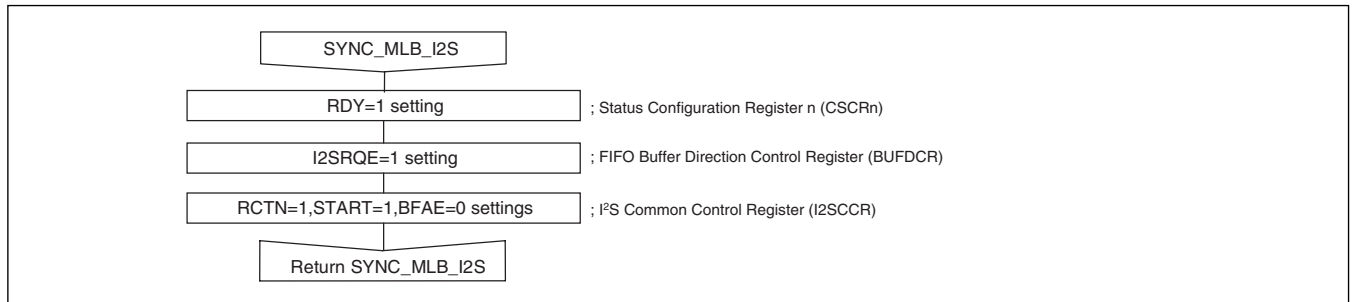
■ Examples of operating procedures

(1) Initial settings

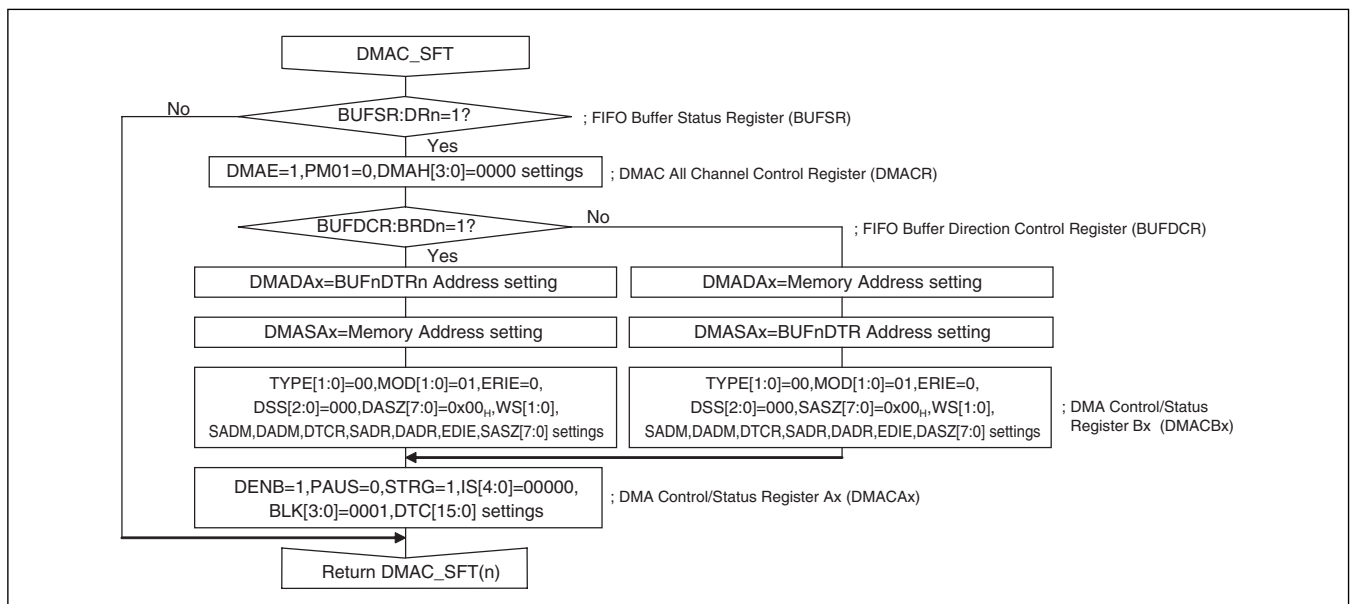
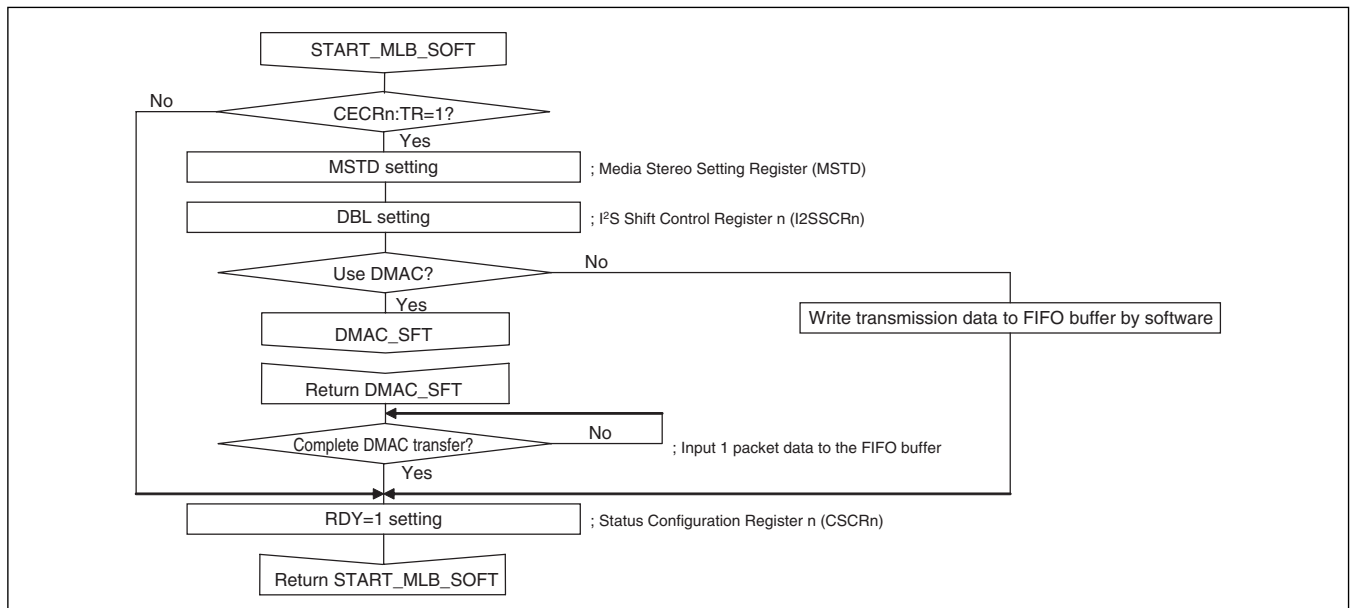


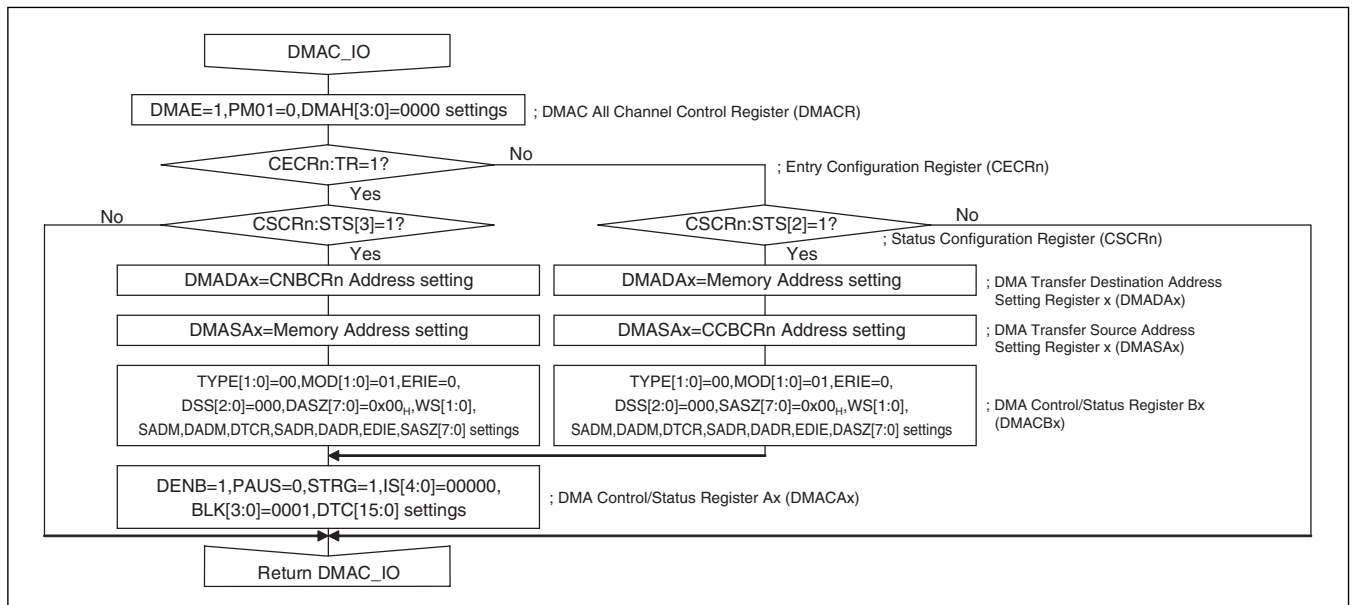
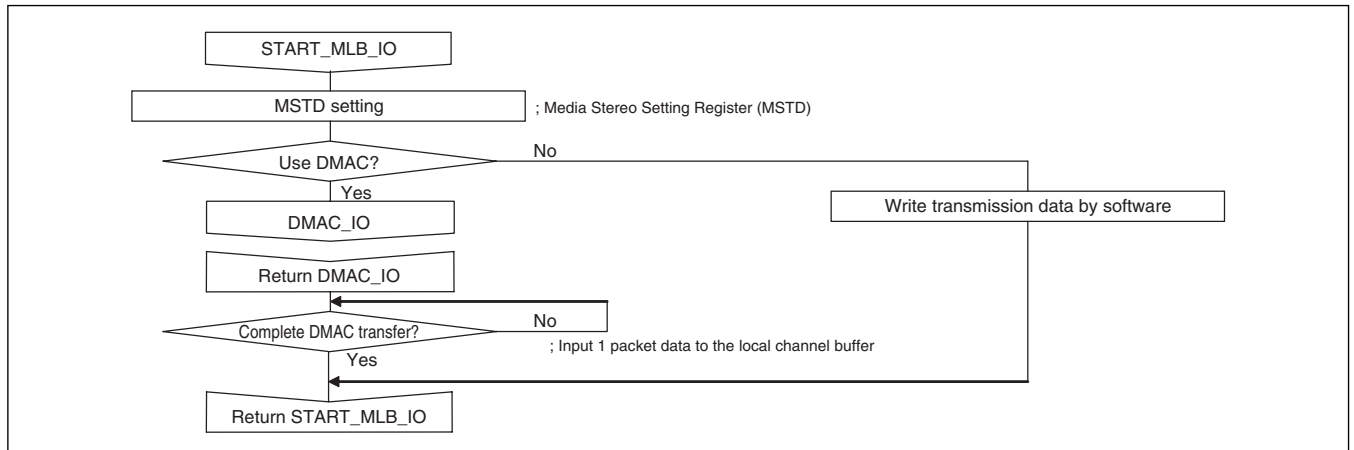


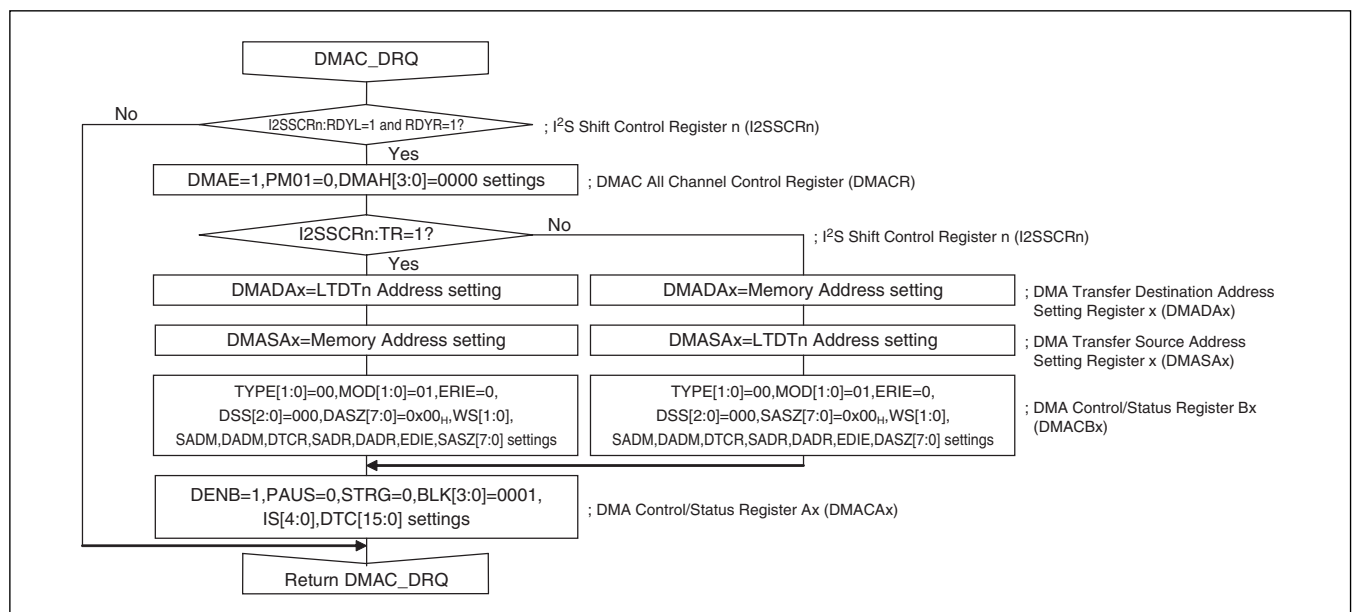
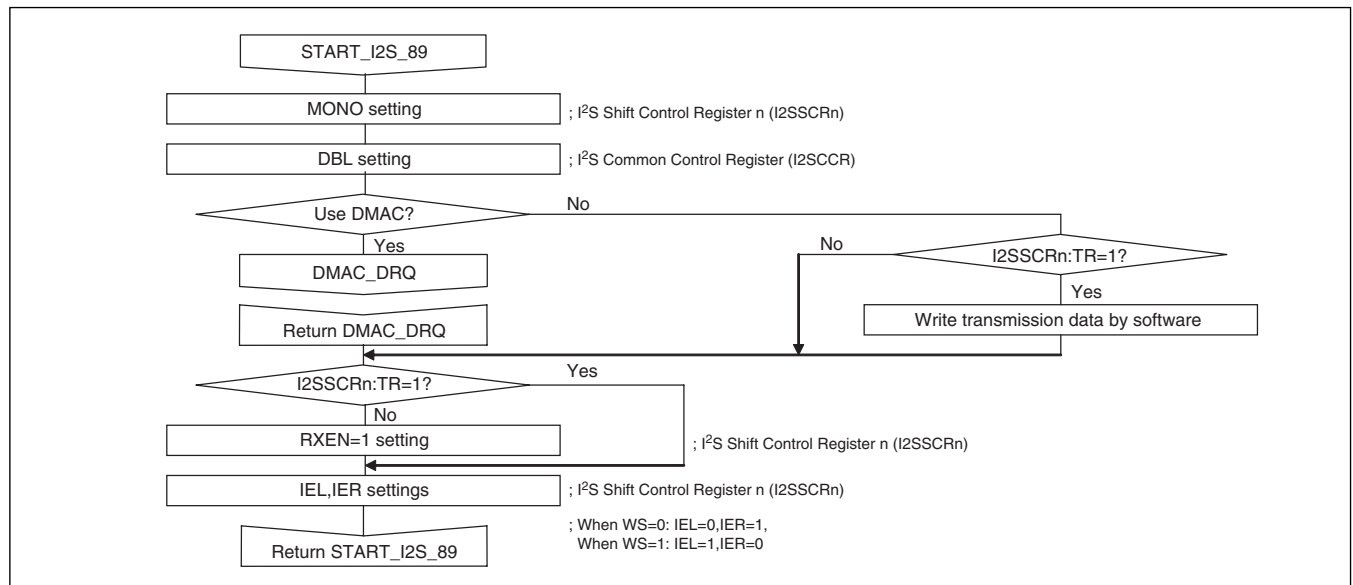


(2) Activation for the case where sync data is flowed between MediaLB and I²S (using circular buffering)

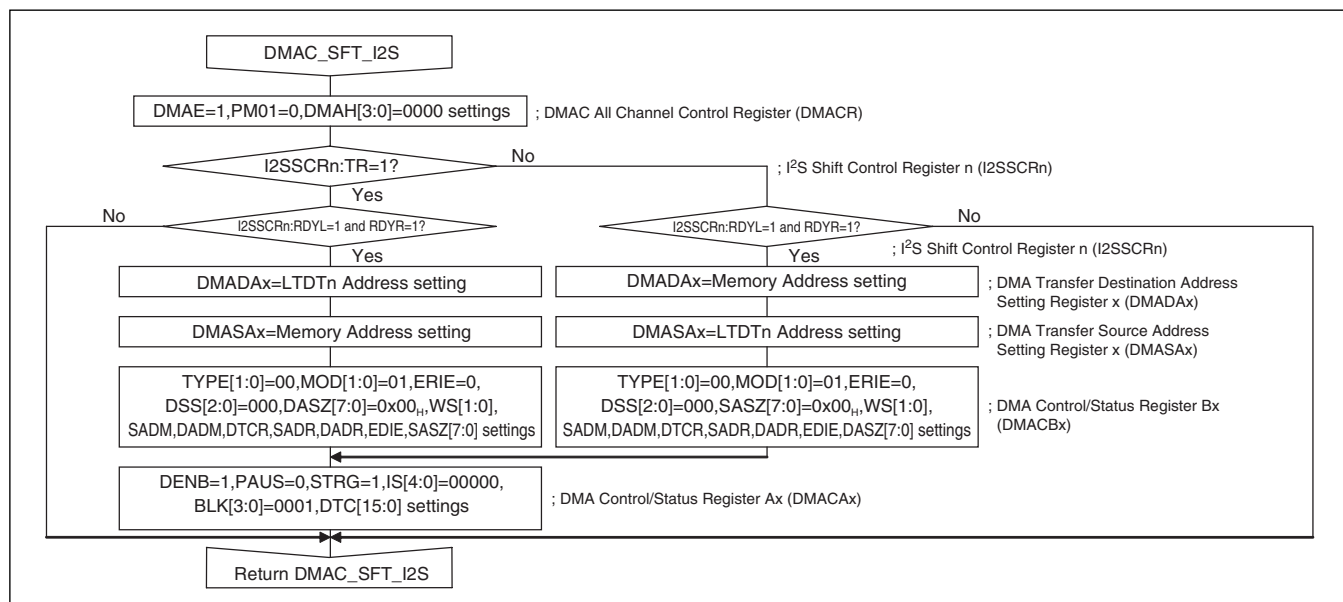
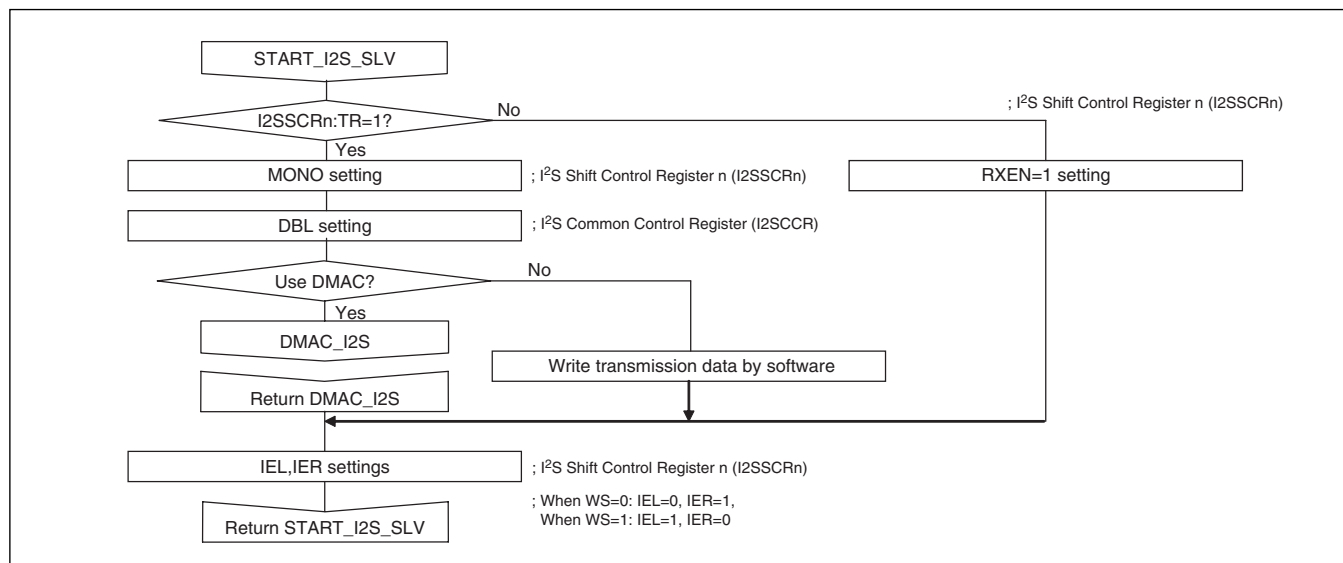
(3) Activation for the case where data is exchanged between MediaLB and software (using ping-pong buffering)



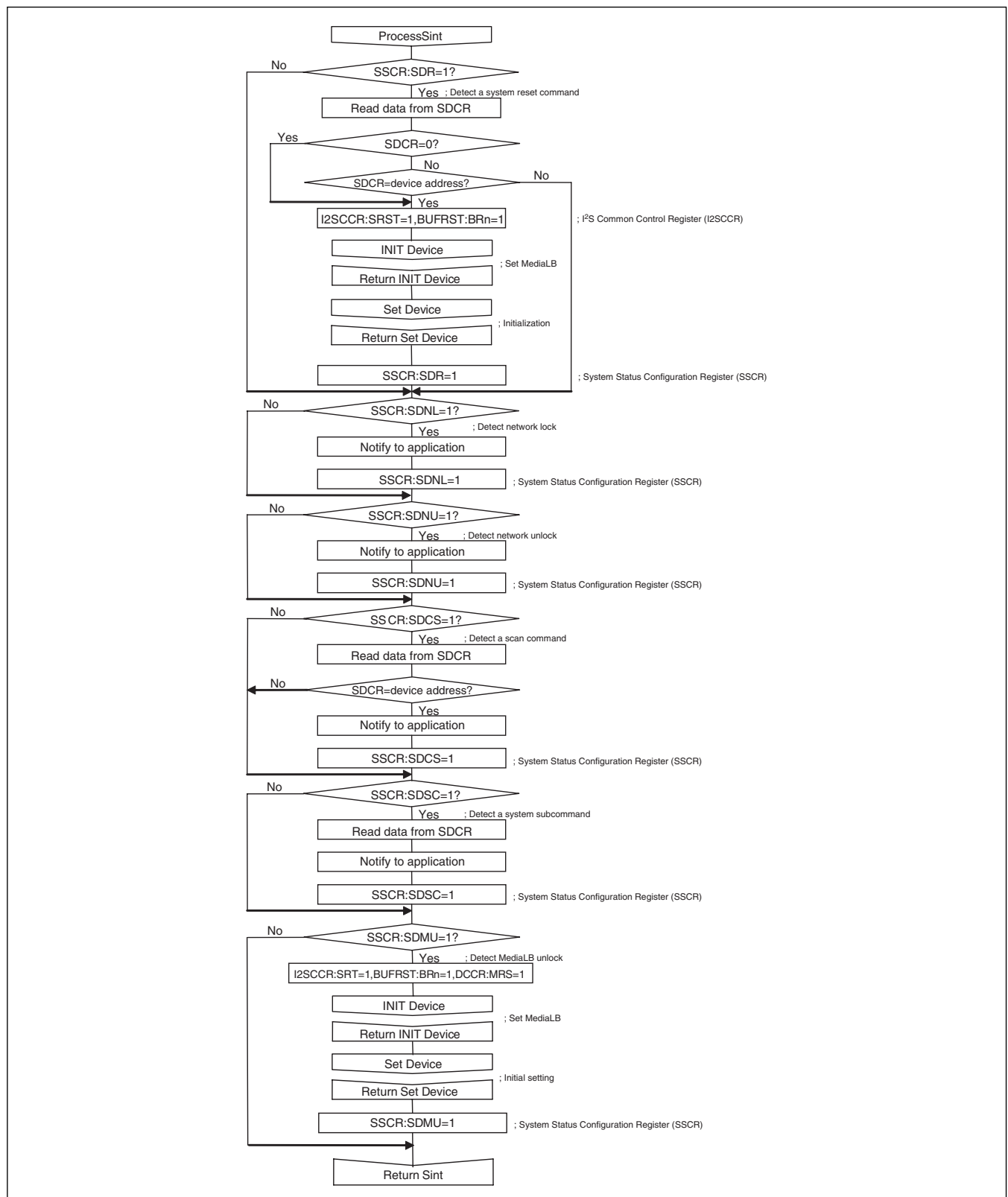
MB91460 Series**(4) Case of transmission from MediaLB to INIC (using IO mode)**

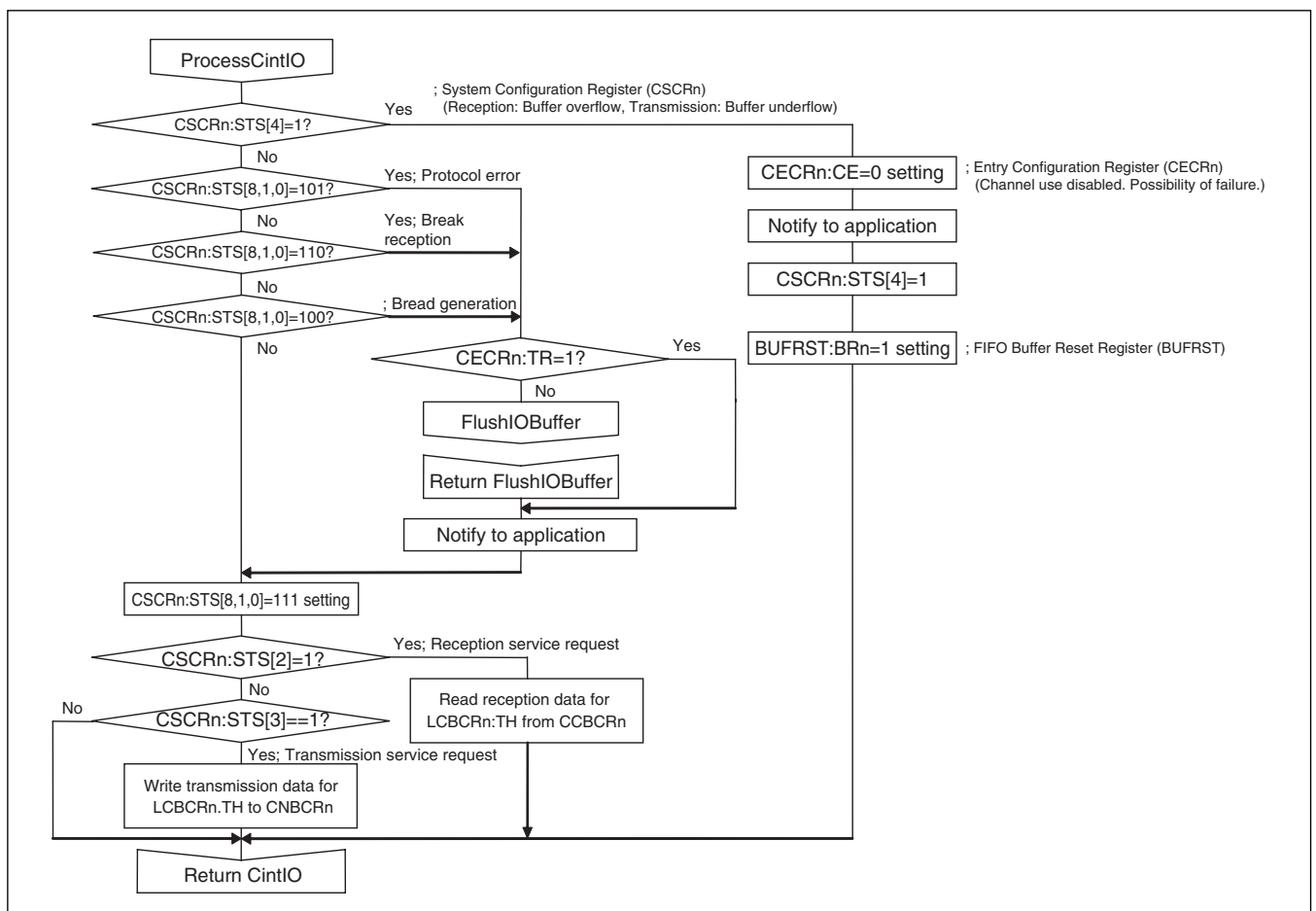
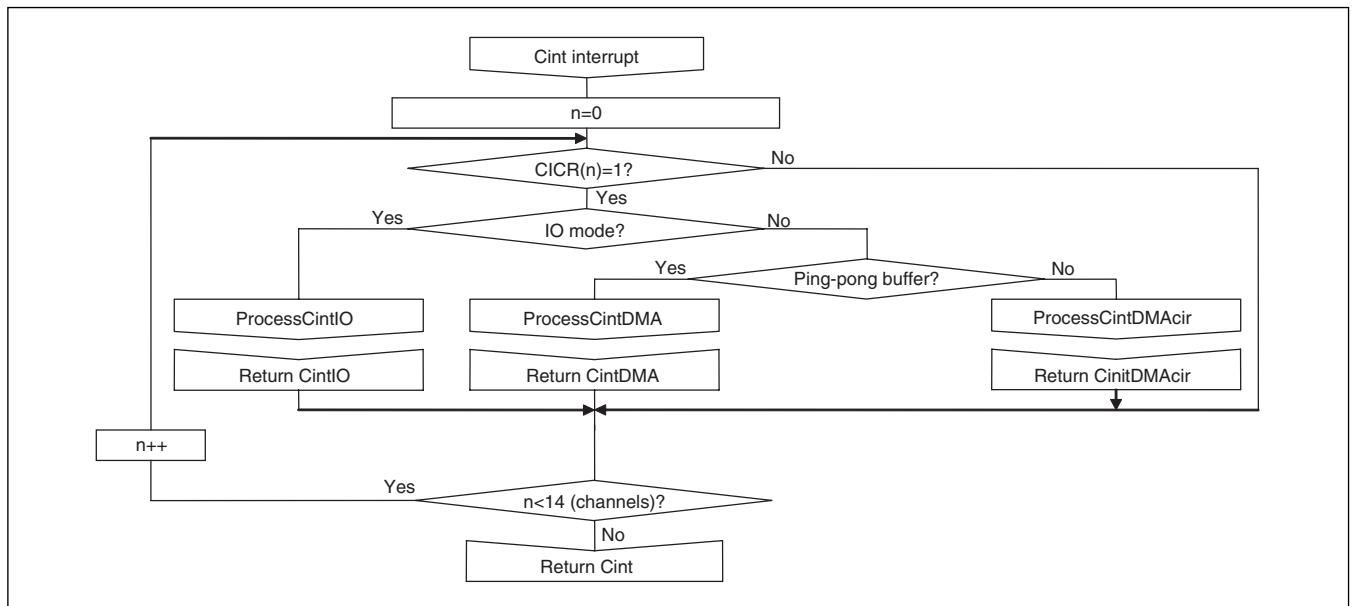
(5) Case of transmission/reception of sync data from I²S channels 8 and 9

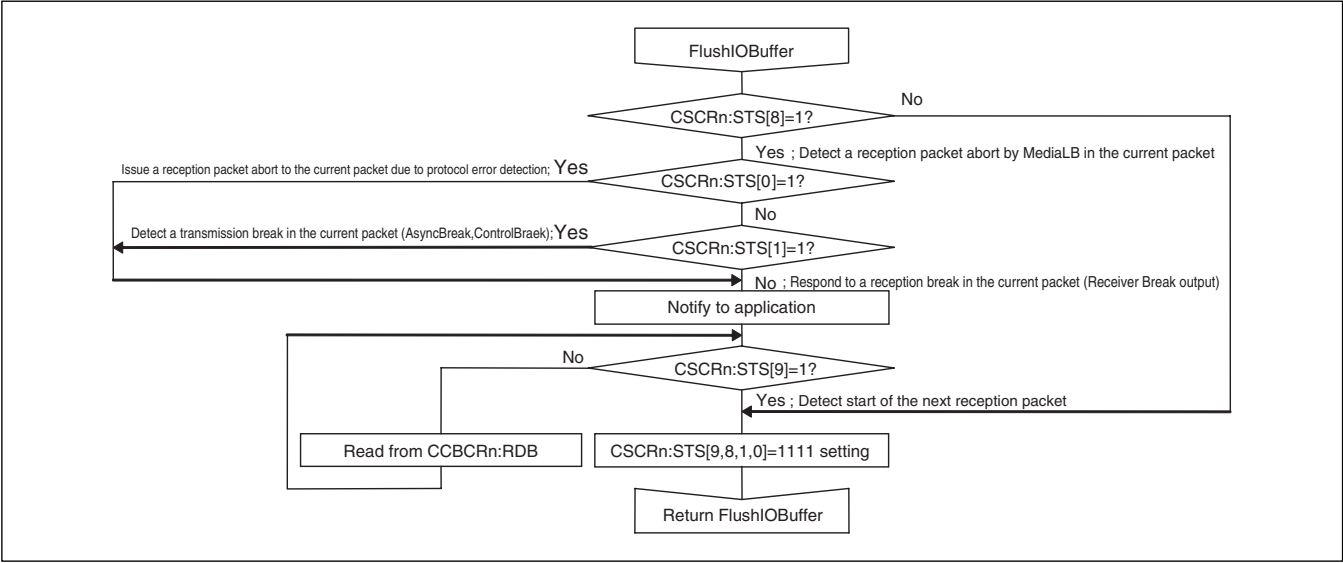
(6) Case where I^2S is activated as slave



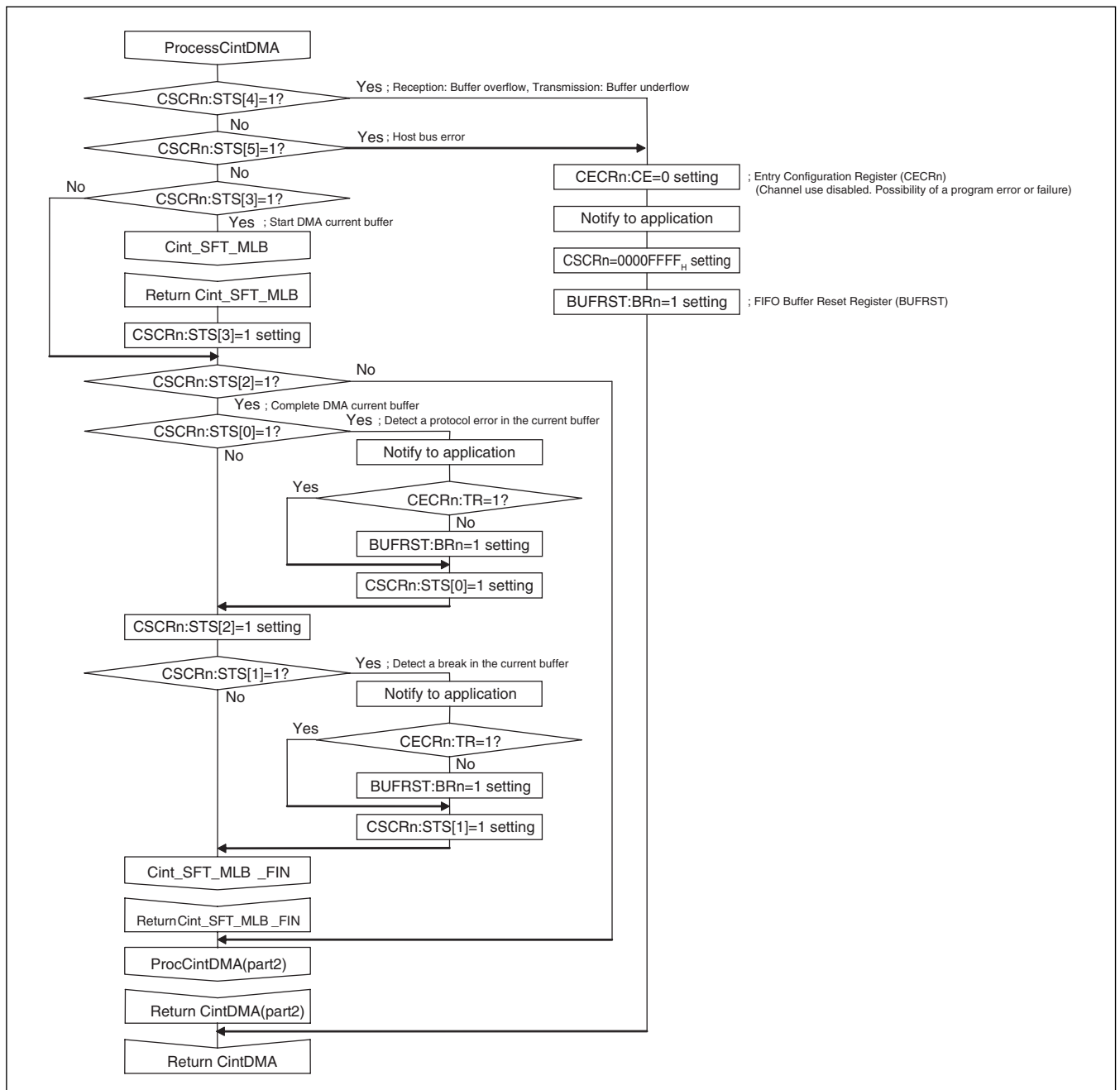
(7) System interrupts

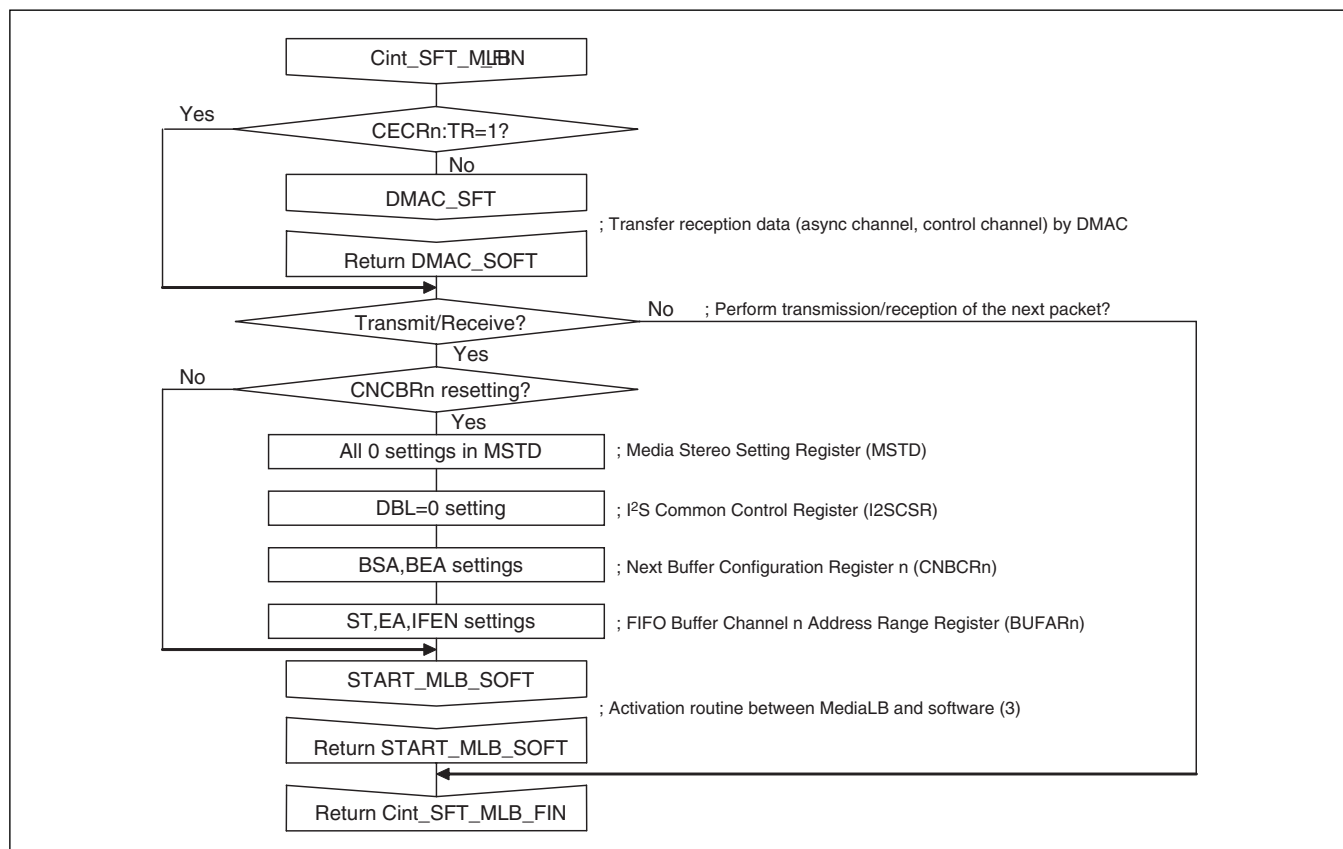
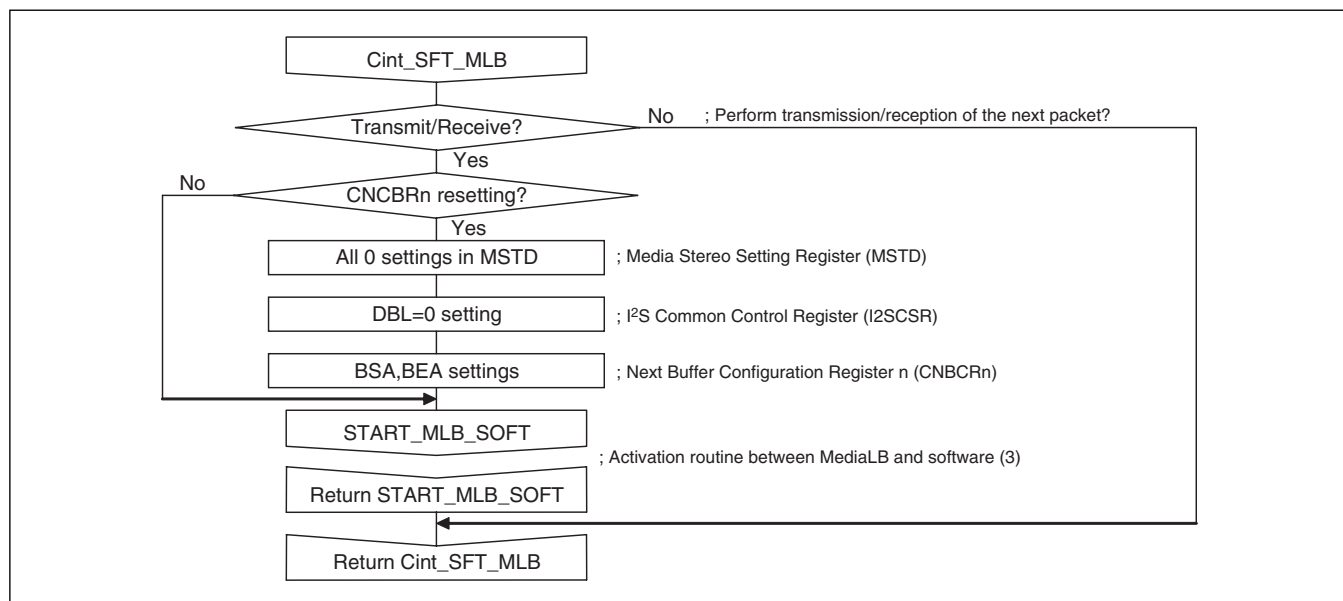


MB91460 Series**(8) Channel interrupts**

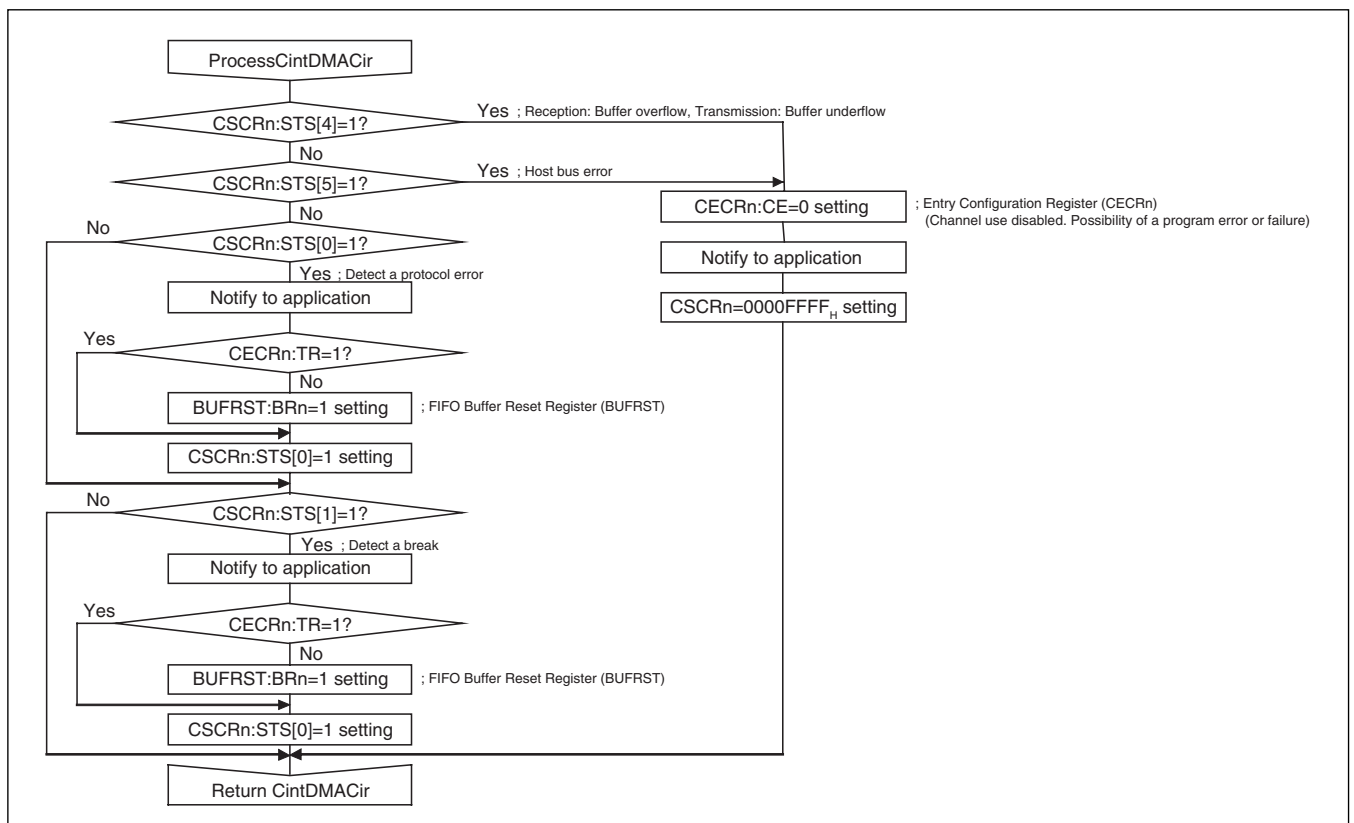
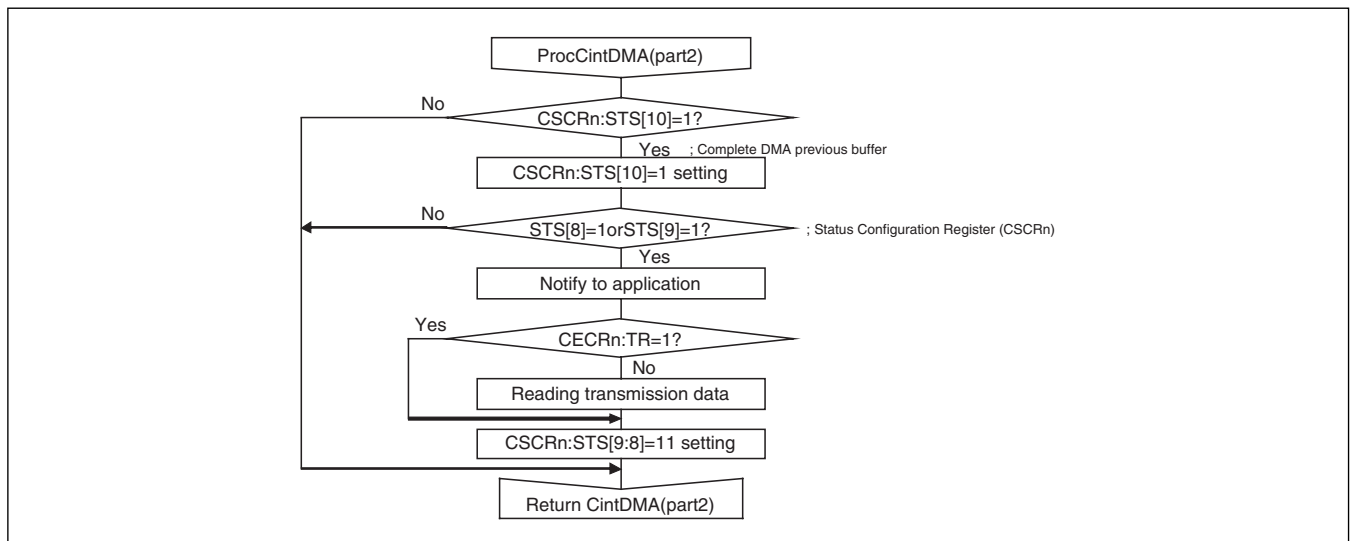


MB91460 Series





MB91460 Series



MB91460 Series**Chapter 61 I²S (Inter-Integrated Circuit Sound)**

This chapter describes the I²S features.

61.1. Overview of the I²S

The I²S is a serial data communication interface developed for digital audio systems. It has 32-byte buffers.

■ I²S features

		Function
1	Channels	10 channels (n=0 to 9)
2	Data buffer	<ul style="list-style-type: none"> • Right and left data registers • Transmission/Reception buffer 32 bits × 8 stages (each channel)
3	Transfer format	MSB first
4	Baud rate	A dedicated baud rate generator is provided
5	Data length	Variable between 3 and 16 bits
6	Error detection	Overrun error, underrun error
7	Master/Slave	Supports master/slave

Note) In master, only transmission is enabled in ch.8 and ch.9. In slave, transmission and reception settings are enabled in ch.8 and ch.9.

61.2. I²S Registers

A list of the I²S registers is shown below.

Table 61.2-1 List of the I²S Registers (1 / 3)

Address	bit31	bit16	bit15	bit8	bit7	bit0
006310 _H	I ² S Common Control Register (I2SCCR)			I ² S Rate Setting Register (I2SRSR)		
Channel 0 register						
006318 _H	I ² S Shift Control Register 0 (I2SSCR0)			Reserved		
00631C _H	Left Data Register 0 (LTDT0)			Right Data Register 0 (RTDT0)		
Channel 1 register						
006320 _H	I ² S Shift Control Register 1 (I2SSCR1)			Reserved		
006324 _H	Left Data Register 1 (LTDT1)			Right Data Register 1 (RTDT1)		
Channel 2 register						
006328 _H	I ² S Shift Control Register 2 (I2SSCR2)			Reserved		
00632C _H	Left Data Register 2 (LTDT2)			Right Data Register 2 (RTDT2)		

Table 61.2-1 List of the I²S Registers (2 / 3)

Address	bit31	bit16	bit15	bit8	bit7	bit0
Channel 3 register						
006330 _H	I ² S Shift Control Register 3 (I2SSCR3)			Reserved		
006334 _H	Left Data Register 3 (LTDT3)			Right Data Register 3 (RTDT3)		
Channel 4 register						
006338 _H	I ² S Shift Control Register 4 (I2SSCR4)			Reserved		
00633C _H	Left Data Register 4 (LTDT4)			Right Data Register 4 (RTDT4)		
Channel 5 register						
006340 _H	I ² S Shift Control Register 5 (I2SSCR5)			Reserved		
006344 _H	Left Data Register 5 (LTDT5)			Right Data Register 5 (RTDT5)		
Channel 6 register						
006348 _H	I ² S Shift Control Register 6 (I2SSCR6)			Reserved		
00634C _H	Left Data Register 6 (LTDT6)			Right Data Register 6 (RTDT6)		
Channel 7 Register						
006350 _H	I ² S Shift Control Register 7 (I2SSCR7)			Reserved		
006354 _H	Left Data Register 7 (LTDT7)			Right Data Register 7 (RTDT7)		

Table 61.2-1 List of the I²S Registers (3 / 3)

Address	bit31	bit16	bit15	bit8	bit7	bit0
Channel 8 Register						
006358 _H	I ² S Shift Control Register 8 (I2SSCR8)			I ² S Transfer Count Register 8 (I2SBT8)		I ² S Buffer Control Register 8 (I2SBCR8)
00635C _H	Left Data Register 8 (LTDT8)			Right Data Register 8 (RTDT8)		
Channel 9 register						
006360 _H	I ² S Shift Control Register 9 (I2SSCR9)			I ² S Transfer Count Register 9 (I2SBT9)		I ² S Buffer Control Register 9 (I2SBCR9)
006364 _H	Left Data Register 9 (LTDT9)			Right Data Register 9 (RTDT9)		

■ I²S bit allocations

Table 61.2-2 I²S Bit Allocations

- Upper 16 bits

	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
I2SSCCR	-	TXEN	RCTN	BFAE	SRST	STUS	DBL	START	-					MAS	WL	WS
I2SSCRn	-				MONO	IEE	ERR	RXEN	TR	-	RINIT	-	IEL	IER	RDYL	RDYR
LTDTn	LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0

- Lower 16 bits

	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit09	Bit08	Bit07	Bit06	Bit05	Bit04	Bit03	Bit02	Bit01	Bit00
I2SRSR	-								-		PS1	PS0	BL3	BL2	BL1	BL0
I2SBTn/ I2SBCRn (ch.8/ch.9 only)	-				BTD3	BTD2	BTD1	BTD0	-			BTC3	BTC2	BTC1	BTC0	BEN
RTDTn	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

61.2.1 I²S Common Control Register (I2SSCCR)

The I²S common control register sets software reset, WS start, prescaler, bit length, I²S clock output start, PCM data double transfer, and indicates the clock output status.

■ I²S Common Control Register (I2SSCCR)

Figure 61.2-1 shows the bit configuration of the I²S common control register and Table 61.2-3 shows the functions of the bits.

Figure 61.2-1 Bit Configuration of the I²S Common Control Register (I2SCCR)

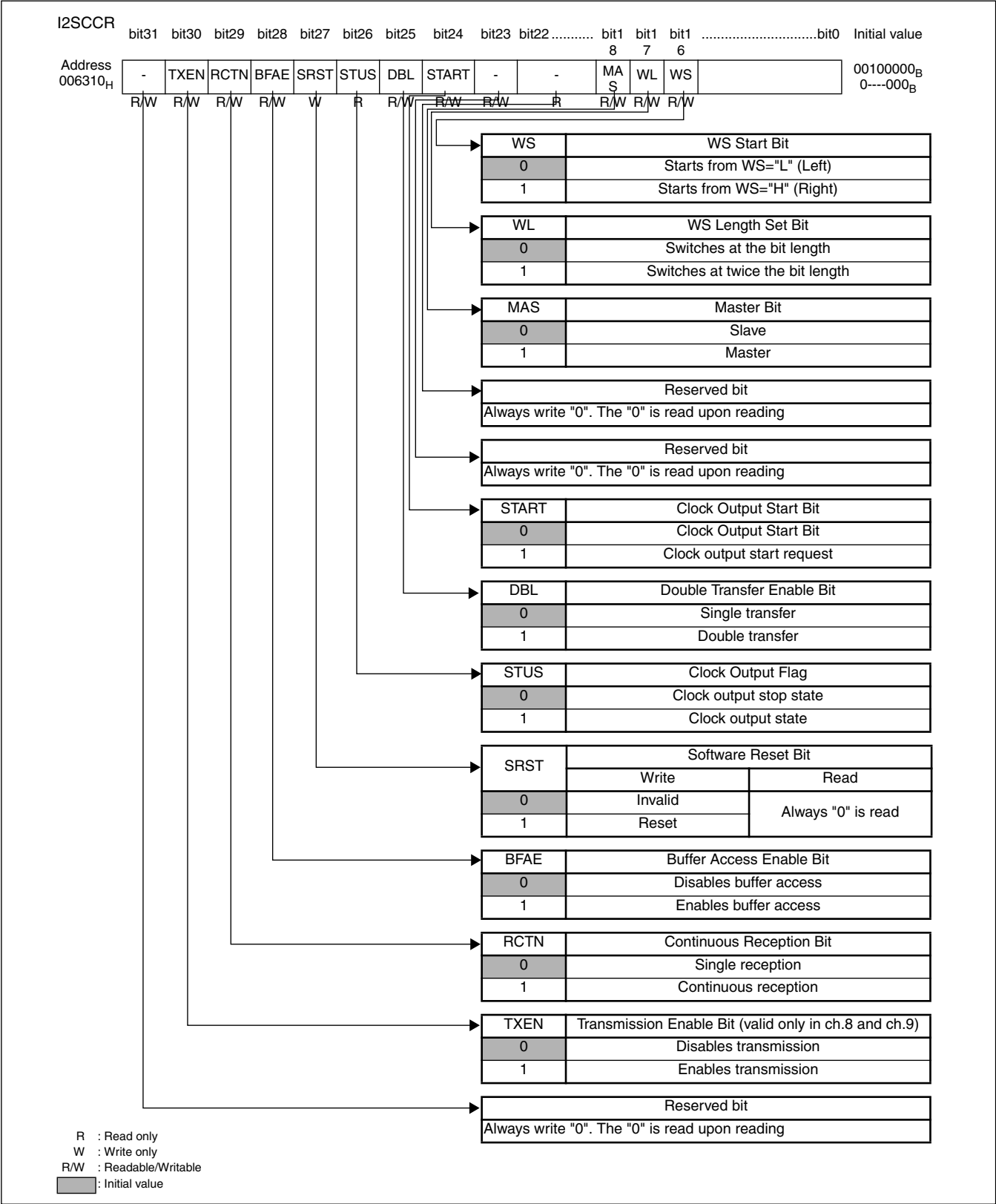


Table 61.2-3 Functional Descriptions on the I²S Common Control Register Bits (1 / 2)

Bit Name		Function
bit31	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit30	TXEN: Transmission Enable Bit	<p>Transmission enable bit.</p> <p>Set to "0": Disables transmission.</p> <p>Set to "1": Enables transmission.</p> <p>When I2SCCR:TXEN="0", transmission data is written to the right and left registers of ch.8 and ch.9. When setting I2SCCR:TXEN="1", transmission data of ch.8 and ch.9 is output on the I²S bus simultaneously. If I2SCCR:TXEN="1" is changed to "0", the transmission is disabled after transmission data output is completed.</p> <p><Notes></p> <ul style="list-style-type: none"> Valid only in ch.8 and ch.9. In other channels, after transmission data is written to the right and left registers, it is transmitted when I2SSCRn:START=1, regardless of this bit. Make sure I2SSCRn:TR is set to "1".
bit29	RCTN: Continuous Reception Bit	<p>Continuous reception bit. Once reception is activated, the reception is performed continuously. Setting this bit to "0" stops continuous reception.</p> <p>Set to "0": Reception is performed every reception activation.</p> <p>Set to "1": Reception is performed continuously.</p> <p>If this bit is set to "1" and I2SSCRn:TR bit is set to "1", underrun error detection is enabled.</p>
bit28	BFAE: Buffer Access Enable Bit	<p>Enables writing "1" to I2SCCR:START from the buffer.</p> <p>Set to "0": Disables writing "1" to I2SCCR:START bit from the buffer.</p> <p>Set to "1": Enables writing "1" to I2SCCR:START bit from the buffer.</p> <p><Note></p> <p>Valid only in master mode.</p>
bit27	SRST: Software Reset Bit	<p>Software reset bit.</p> <p>Setting this bit to "1" initializes the internal of the I²S, and I2SCCR:STUS, I2SCCR:START, I2SSCRn:ERR, I2SSCRn:RXEN, I2SSCRn:RDYL, I2SSCRn:RDYR and I2SBTn:BDT[3:0]. Setting this bit to "1" during clock output stops the clock. The "0" is always read upon reading. Setting this bit to "0" has no effect.</p> <p><Notes></p> <p>For I2SSCRn:RDYL and I2SSCRn:RDYR, the initialized value varies depending on the I2SSCRn:TR bit setting.</p> <ul style="list-style-type: none"> If I2SSCRn:TR="1": I2SSCRn:RDYL and I2SSCRn:RDYR are initialized to "1". If I2SSCRn:TR="0": I2SSCRn:RDYL and I2SSCRn:RDYR are initialized to "0".
bit26	STUS: Clock Output Flag	<p>Indicates the output status of the serial clock (ISCK).</p> <ul style="list-style-type: none"> Condition to become "1" <ul style="list-style-type: none"> Writing "1" to I2SCCR:START bit Conditions to become "0" <ul style="list-style-type: none"> Writing "1" to I2SCCR:SRST bit In the case where there is no request for next clock output after the clock is output for 2 words (right and left). <p><Note></p> <p>Valid only in master mode.</p>

Table 61.2-3 Functional Descriptions on the I²S Common Control Register Bits (2 / 2)

Bit Name		Function
bit25	DBL: Double Transfer Enable Bit	<p>Sets whether to transfer the same data twice. When I2SSCRn:TR="1", channels 8 and 9 are enabled. Set to "0": No data addition. Set to "1": The data set previously is written again to the transmission buffer.</p> <p><Note> This bit is valid only for channels 8 and 9. For the MediaLB, this bit is valid only for channels 8 and 9.</p>
bit24	START: Clock Output Start Bit	<p>Determines whether to start clock output. The MediaLB or software sets this bit to "1". Setting this bit to "1" requests clock output, and this bit becomes "0" as the clock output starts. However, when I2SSCCR:RCTN="1" (continuous reception), this bit will not be reset to "0". Setting I2SSCCR:SRST to "1" sets this bit to "0". When an unlock is detected by the MediaLB, I2SSCCR:START is cleared to "0". Writing "0" has no effect.</p> <p><Notes></p> <ul style="list-style-type: none"> For transmission, transmission will not be performed until 32-bit data is written to the register. Transmission is performed when I2SSCCR:START is set to "1" and 32-bit data is written to the register. For a read-modify-write instruction, the "0" is read from this bit. This bit is valid only in master mode.
bit23-bit19	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit18	MAS: Master Bit	<p>Sets master or slave. Set to "0": Sets to slave. Clock is input from ISCK. Set to "1": Sets to master. Clock is output from ISCK. This bit controls the directions of ISCK pin and WS pin. If "1" is set, ISCK pin and WS pin are set to the output direction. If "0" is set, ISCK pin and WS pin are set to the input direction.</p> <p><Notes></p> <ul style="list-style-type: none"> If set to I2SSCCR:MAS="0", accessing to the right and left registers from the MediaLB is prohibited. (Setting DMA mode in the MediaLB is prohibited.) If switching between from master to slave during operation, follow the procedure below: <ul style="list-style-type: none"> (1) Set I2SSCCR:RCTN="0" and I2SSCCR.BFAE="0". (2) Wait until I2SSCCR:STUS="0", then set I2SSCCR:MAS="0".
bit17	WL: WS Length Set Bit	<p>Sets the length of WS to be inverted. Set to "0": The length of WS inversion is the same length as I2SSCCR:BL[3:0]. Set to "1": The length of WS inversion is twice the length of I2SSCCR:BL[3:0].</p> <p><Notes> If set to I2SSCCR:WL="1":</p> <ul style="list-style-type: none"> In reception, the last half of data is discarded. In transmission, the "L" is output for the last half of data. <p>Valid only in master mode.</p>
bit16	WS: WS Start Bit	<p>Determines the level of WS pin at the time of starting communication. Set to "0": Starts from WS="L". Set to "1": Starts from WS="H".</p> <p><Note> If this bit is set to "1", the SDn pin with transmission setting becomes "L".</p>

MB91460 Series

61.2.2 I²S Rate Setting Register(I2SRSR)

The I²S rate setting register sets prescaler and bit length.

■ I²S Rate Setting Register (I2SRSR)

Figure 61.2-2 shows the bit configuration of the I²S rate setting register and Table 61.2-4 shows the functions of the bits.

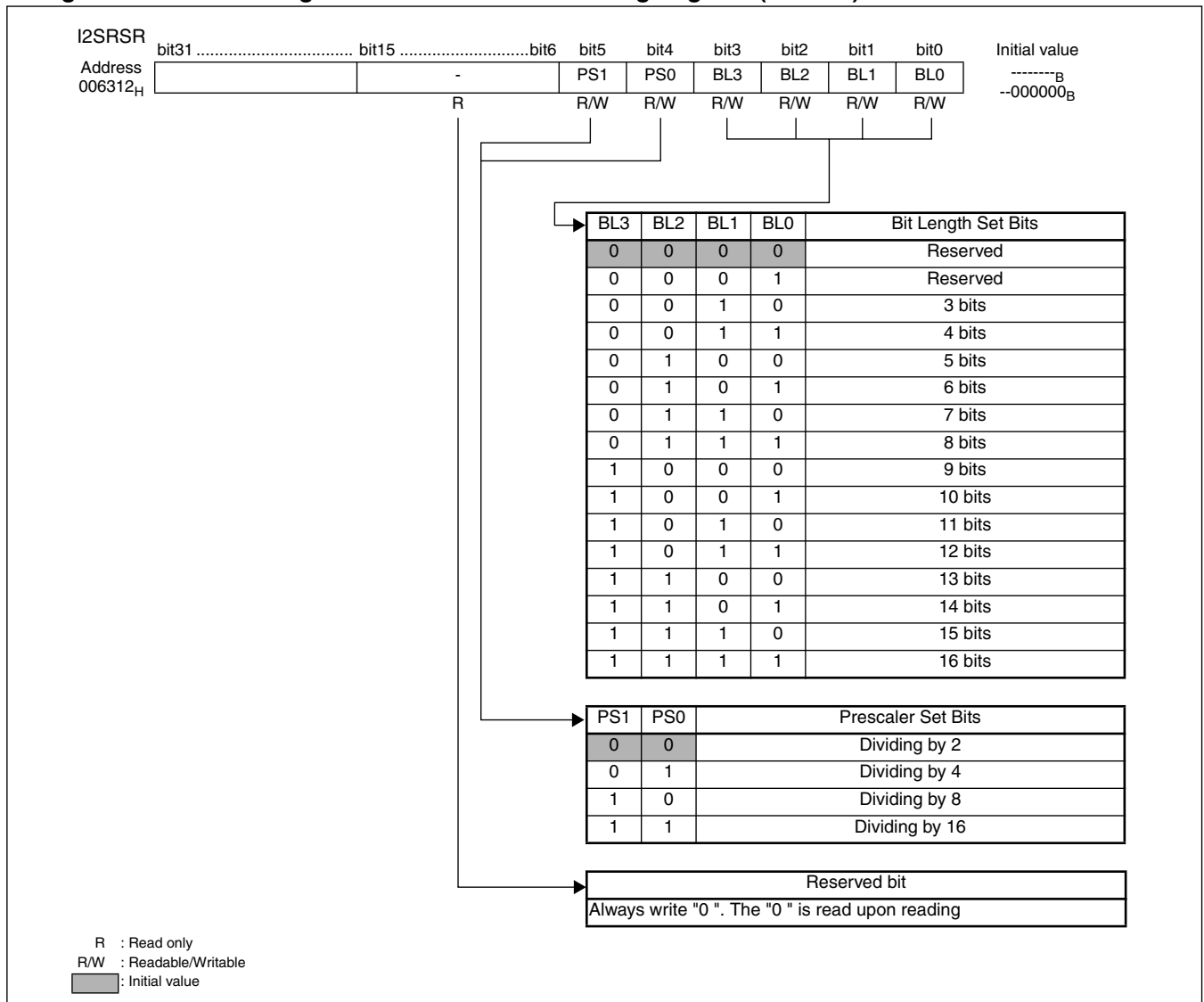
Figure 61.2-2 Bit Configuration of the I²S Rate Setting Register (I2SRSR)

Table 61.2-4 Functional Descriptions on the I²S Rate Setting Register Bits

Bit Name		Function																																																																																					
bit5, bit4	PS1, PS0: Prescaler Set Bits	Sets the division ratio of MLBCLK clock. This setting determines the transfer rate. <table><tr><th>PS1</th><th>PS0</th><th>Division Ratio</th></tr><tr><td>0</td><td>0</td><td>Dividing by 2</td></tr><tr><td>0</td><td>1</td><td>Dividing by 4</td></tr><tr><td>1</td><td>0</td><td>Dividing by 8</td></tr><tr><td>1</td><td>1</td><td>Dividing by 16</td></tr></table>	PS1	PS0	Division Ratio	0	0	Dividing by 2	0	1	Dividing by 4	1	0	Dividing by 8	1	1	Dividing by 16																																																																						
		PS1	PS0	Division Ratio																																																																																			
		0	0	Dividing by 2																																																																																			
		0	1	Dividing by 4																																																																																			
		1	0	Dividing by 8																																																																																			
		1	1	Dividing by 16																																																																																			
<Note> Valid only in master mode.																																																																																							
bit3- bit0	BL3, BL2, BL1, BL0: Bit Length Set Bits	Sets the length of bits. Left data and right data will have the same bit length. <table><tr><th>BL3</th><th>BL2</th><th>BL1</th><th>BL0</th><th>Bit Length</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Reserved</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Reserved</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>3 bits</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>4 bits</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>5 bits</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>6 bits</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>7 bits</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8 bits</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>9 bits</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>10 bits</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>11 bits</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>12 bits</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>13 bits</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>14 bits</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>15 bits</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>16 bits</td></tr></table>	BL3	BL2	BL1	BL0	Bit Length	0	0	0	0	Reserved	0	0	0	1	Reserved	0	0	1	0	3 bits	0	0	1	1	4 bits	0	1	0	0	5 bits	0	1	0	1	6 bits	0	1	1	0	7 bits	0	1	1	1	8 bits	1	0	0	0	9 bits	1	0	0	1	10 bits	1	0	1	0	11 bits	1	0	1	1	12 bits	1	1	0	0	13 bits	1	1	0	1	14 bits	1	1	1	0	15 bits	1	1	1	1	16 bits
		BL3	BL2	BL1	BL0	Bit Length																																																																																	
		0	0	0	0	Reserved																																																																																	
		0	0	0	1	Reserved																																																																																	
		0	0	1	0	3 bits																																																																																	
		0	0	1	1	4 bits																																																																																	
		0	1	0	0	5 bits																																																																																	
		0	1	0	1	6 bits																																																																																	
		0	1	1	0	7 bits																																																																																	
		0	1	1	1	8 bits																																																																																	
		1	0	0	0	9 bits																																																																																	
		1	0	0	1	10 bits																																																																																	
		1	0	1	0	11 bits																																																																																	
		1	0	1	1	12 bits																																																																																	
		1	1	0	0	13 bits																																																																																	
		1	1	0	1	14 bits																																																																																	
		1	1	1	0	15 bits																																																																																	
		1	1	1	1	16 bits																																																																																	
		Bits are transmitted/received from the upper bit of the right and left data registers according to this setting. When receiving, the remaining bits are set to 0.																																																																																					
		<Note> When using channels 0 to 7, since it takes 8 clocks per channel to access a buffer, an overrun or underrun may occur during transfer to the buffer because of the bit length, division and WS length. Make sure the bit length, etc. is set so as to satisfy the following equation.																																																																																					
• WL=0 Division ratio ≠ bit length 8 ≠ number of channels																																																																																							
• WL=1 Division ratio ≠ bit length 4 ≠ number of channels																																																																																							

61.2.3 I²S Shift Control Register n (I2SSCRn)

The I²S shift control register (I2SSCRn) performs stereo/monaural settings, enabling interrupts, error flag checks, reception enabling setting, transmission/reception settings, initialization of status flags for right and left registers and status checks for right and left registers.

■ I²S Shift Control Register n (I2SSCRn)

Figure 61.2-3 shows the bit configuration of the I²S shift control register n (I2SSCRn) and Table 61.2-5 shows the functions of the bits.

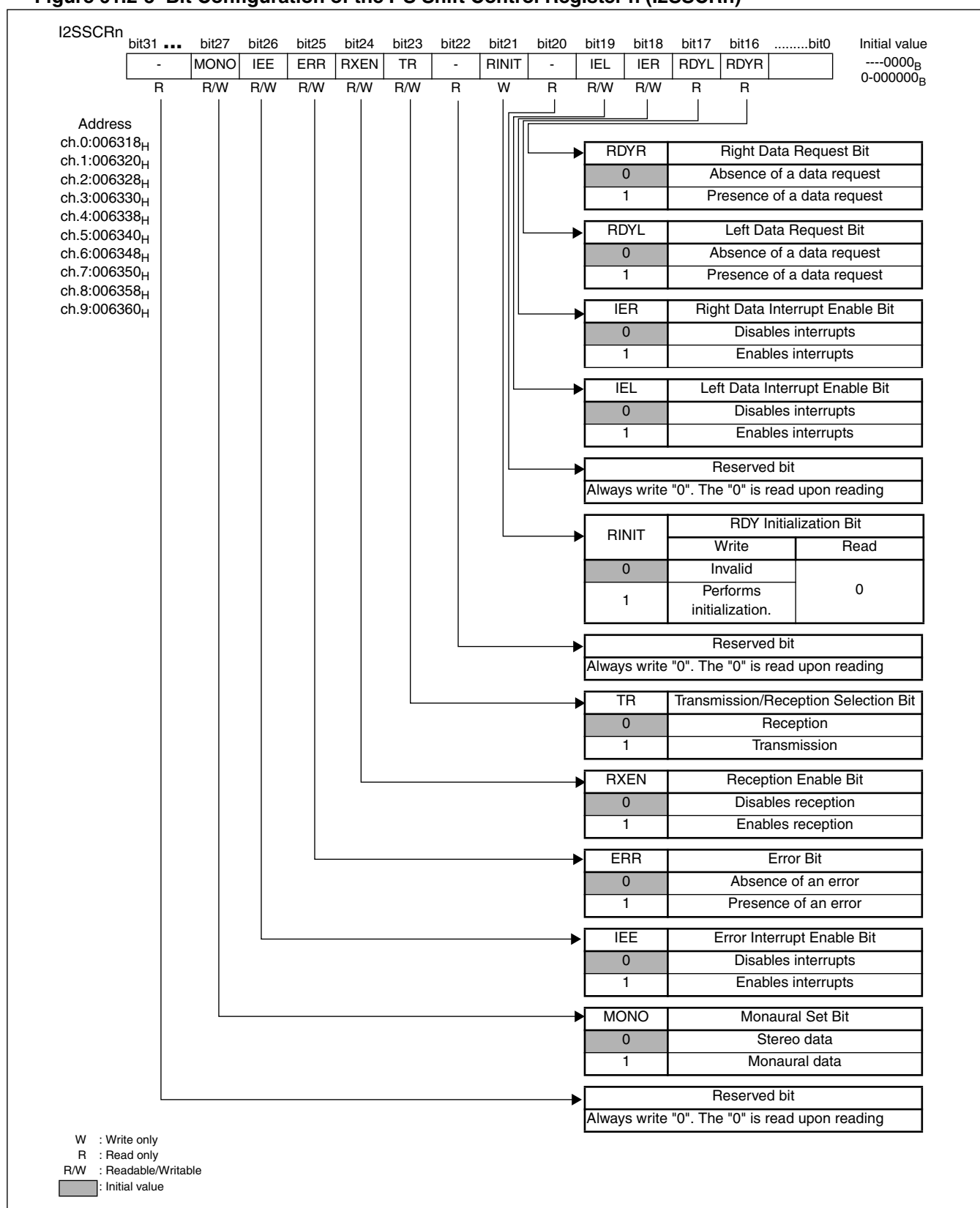
Figure 61.2-3 Bit Configuration of the I²S Shift Control Register n (I2SSCRn)

Table 61.2-5 Functional Descriptions on the I²S Shift Control Register n (I2SSCRn) Bits (1 / 3)

Bit Name		Function
bit31-bit28	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit27	MONO: Monaural Set Bit	Handles data as monaural data if the data is written from software (a write via the PB bus). If "1" is set, when data is written to the right register, 0 is written to the left register, and when data is written to the left register, 0 is written to the right register. If "0" is set, data is written to the right and left registers and the data is transmitted as stereo data. <Notes> <ul style="list-style-type: none"> - This bit is valid only when I2SSCRn:TR=1. - If this bit is set to "1", make sure to write 16-bit data to either the right data register or left data register. - Make sure this bit is set to "0" except for ch.8 and ch.9.
bit26	IEE: Error Interrupt Enable Bit	Enables interrupts when I2SSCRn:ERR becomes "1". Set to "0": Disables interrupts by I2SSCRn:ERR bit. Set to "1": Enables interrupts by I2SSCRn:ERR bit.
bit25	ERR: Error Bit	Indicates the detection of an error. Writing "0" while this bit is set to "1" clears this bit to "0", however, if "1" is written, the previous state is maintained. The conditions in which this bit is set to "1" are as follows: <ul style="list-style-type: none"> - In the case where an overrun occurs in reception. - In the case where an underrun occurs in transmission. The condition in which an underrun occurs is as follows: <ul style="list-style-type: none"> - In the case where no next valid data exists in right and left data registers after data is written to the right and left data registers and is output as transmission data while I2SSCCR:RCTN="1". The conditions in which no underrun occurs are as follows: <ul style="list-style-type: none"> - In the case where no writing is performed to right and left data registers. - In the case where no writing is performed to right and left data registers after this bit is cleared to "0". - In the case where I2SSCCR:RCTN="0" If an overrun occurs, written data is discarded. If an underrun occurs, "L" is output from the SDn pin. This bit becomes "0" when "1" is written to I2SSCCR:SRST. <Note> For a read-modify-write instruction, the "1" is read from this bit.
bit24	RXEN: Reception Enable Bit	Sets for enabling reception. When I2SSCRn:TR="0" and this bit is "1", reception is enabled. When I2SSCRn:TR="1", no reception is performed even if this bit is set to "1" because setting "1" has no effect. If this bit is set to "0" in the middle of transmission, the transmission is stopped immediately. This bit becomes "0" when "1" is written to I2SSCCR:SRST. <Note> In master, make sure this bit for ch.8/ch.9 is set to "0" since ch.8 and ch.9 cannot perform reception.

Table 61.2-5 Functional Descriptions on the I²S Shift Control Register n (I2SSCRn) Bits (2 / 3)

Bit Name		Function
bit23	TR: Transmission/ Reception Selection Bit	Transmission/Reception selection bit. Set to "0": Operates as reception. Set to "1": Operates as transmission. If "1" is set, the SDn pin becomes output state. If "0" is set, the SDn pin becomes input state. <Note> For the channels that are not used for transmission/reception, make sure this bit is set for "0" and RXE bit is set to "0".
bit22	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit21	RINIT: RDY Initialization Bit	Initializes I2SSCRn:RDYL and I2SSCRn:RDYR. When writing "1" to this bit, I2SSCRn:RDYL and I2SSCRn:RDYR are set to "0" if I2SSCRn:TR="0", and I2SSCRn:RDYL and I2SSCRn:RDYR are set to "1" if I2SSCRn:TR="1". If I2SBCRn:BEN (buffer enable)="1", writing "1" to this bit clears the buffer transfer count (I2SBTn:BDT) to 0. The "0" is always read upon reading.
bit20	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit19	IEL: Left Data Interrupt Enable Bit	Enables interrupts when I2SSCRn:RDYL becomes "1". Set to "0": Disables interrupts by I2SSCRn:RDYL bit. Set to "1": Enables interrupts by I2SSCRn:RDYL bit.
bit18	IER: Right Data Interrupt Enable Bit	Enables interrupts when I2SSCRn:RDYR becomes "1". Set to "0": Disables interrupts by I2SSCRn:RDYR bit. Set to "1": Enables interrupts by I2SSCRn:RDYR bit.
bit17	RDYL: Left Data Request Bit	Indicates a data request for left. In reception, receiving valid data in the left data register sets this bit to "1". Reading LD7-LD0 of the left data register clears this bit to "0". In transmission, this bit is set to "1" if no valid data exists in the left data register. Writing data to LD7-LD0 of the left data register sets this bit to "0", and transferring the data to the transmission shift register sets this bit to "1". This bit is initialized when "1" is written to I2SSCCR:SRST bit. When I2SBCRn:BEN="1", I2SBTn:BDT[3:0] and this bit are compared, and I2SSCRn:RDYL is kept to "1" while either of the following conditions is met: <ul style="list-style-type: none"> • When I2SSCRn:TR="1"(transmission), I2SSCRn:RDYL is kept to "1" in the state where the number of data written in the buffer is smaller than the number of data set (I2SBTn:BDT[3:0]I2SBCRn:BTC[3:0]) until LTDTn is written. • When I2SSCRn:TR="0"(reception), I2SSCRn:RDYL is kept to "1" in the state where I2SBTn:BDT[3:0]=I2SBCRn:BTC[3:0] and LTDTn is written or the state where the number of data written in the buffer is greater than the number of data set (I2SBTn:BDT[3:0]>I2SBCRn:BTC[3:0]). <Notes> At software reset, the initial value of I2SSCRn:RDYL varies depending on the I2SSCRn:TR bit setting. <ul style="list-style-type: none"> - If I2SSCRn:TR="1": I2SSCRn:RDYL is initialized to "1". - If I2SSCRn:TR="0": I2SSCRn:RDYL is initialized to "0". - When I2SSCCR:DBL="1" and I2SSCRn:TR="1", this bit is not set to "1" even if the first data is transferred to the shift register. This bit is set to "1" when the second data is transferred to the shift register. - If I2SSCRn:MONO="1" is set, this bit becomes "0" when 16-bit data is written to the left data register or right data register.

Table 61.2-5 Functional Descriptions on the I²S Shift Control Register n (I2SSCRn) Bits (3 / 3)

Bit Name		Function
bit16	RDYR: Right Data Request Bit	<p>Indicates a data request for right.</p> <p>In reception, receiving valid data in the right data register sets this bit to "1". Reading the right data register clears this bit to "0".</p> <p>In transmission, this bit is set to "1" if no valid data exists in the right data register. Writing data to the right data register sets this bit to "0", and transferring the data to the transmission shift register sets this bit to "1".</p> <p>This bit is initialized when "1" is written to I2SSCCR:SRST bit.</p> <p>When I2SBCRn:BEN="1", I2SBTn:BDT[3:0] and this bit are compared, and I2SSCRn:RDYR is kept to "1" while either of the following conditions is met.</p> <ul style="list-style-type: none"> When I2SSCRn:TR="1"(transmission), I2SSCRn:RDYR is kept to "1" in the state where the number of data written in the buffer is smaller than the number of data set (I2SBTn:BDT[3:0]I2SBCRn:BTC[3:0]) until RTDTn is written. When I2SSCRn:TR="0"(reception), I2SSCRn:RDYR is kept to "1" in the state where I2SBTn:BDT[3:0]=I2SBCRn:BTC[3:0] and RTDTn is written or the state where the number of data written in the buffer is greater than the number of data set (I2SBTn:BDT[3:0]>I2SBCRn:BTC[3:0]). <p><Notes></p> <p>At software reset, the initial value of I2SSCRn:RDYR varies depending on the I2SSCRn:TR bit setting.</p> <ul style="list-style-type: none"> If I2SSCRn:TR="1": I2SSCRn:RDYR is set to "0" during software reset, and set to "1" after the software reset. If I2SSCRn:TR="0": I2SSCRn:RDYR is initialized to "0". When I2SSCCR:DBL="1" and I2SSCRn:TR="1", this bit is not set to "1" even if the first data is transferred to the shift register. This bit is set to "1" when the second data is transferred to the shift register. If I2SSCRn:MONO="1" is set, this bit becomes "0" when 16-bit data is written to the left data register or right data register.

61.2.4 I²S Buffer Transfer Count Register n (I2SBTn)

The I²S buffer transfer count register indicates the number of valid data in the buffer. Supported only in ch.8 and ch.9.

■ I²S Buffer Transfer Count Register n (I2SBTn)

Figure 61.2-4 shows the bit configuration of the I²S buffer transfer count register and Table 61.2-6 shows the functions of the bits.

Figure 61.2-4 Bit Configuration of the I²S Buffer Transfer Count Register n (I2SBTn)

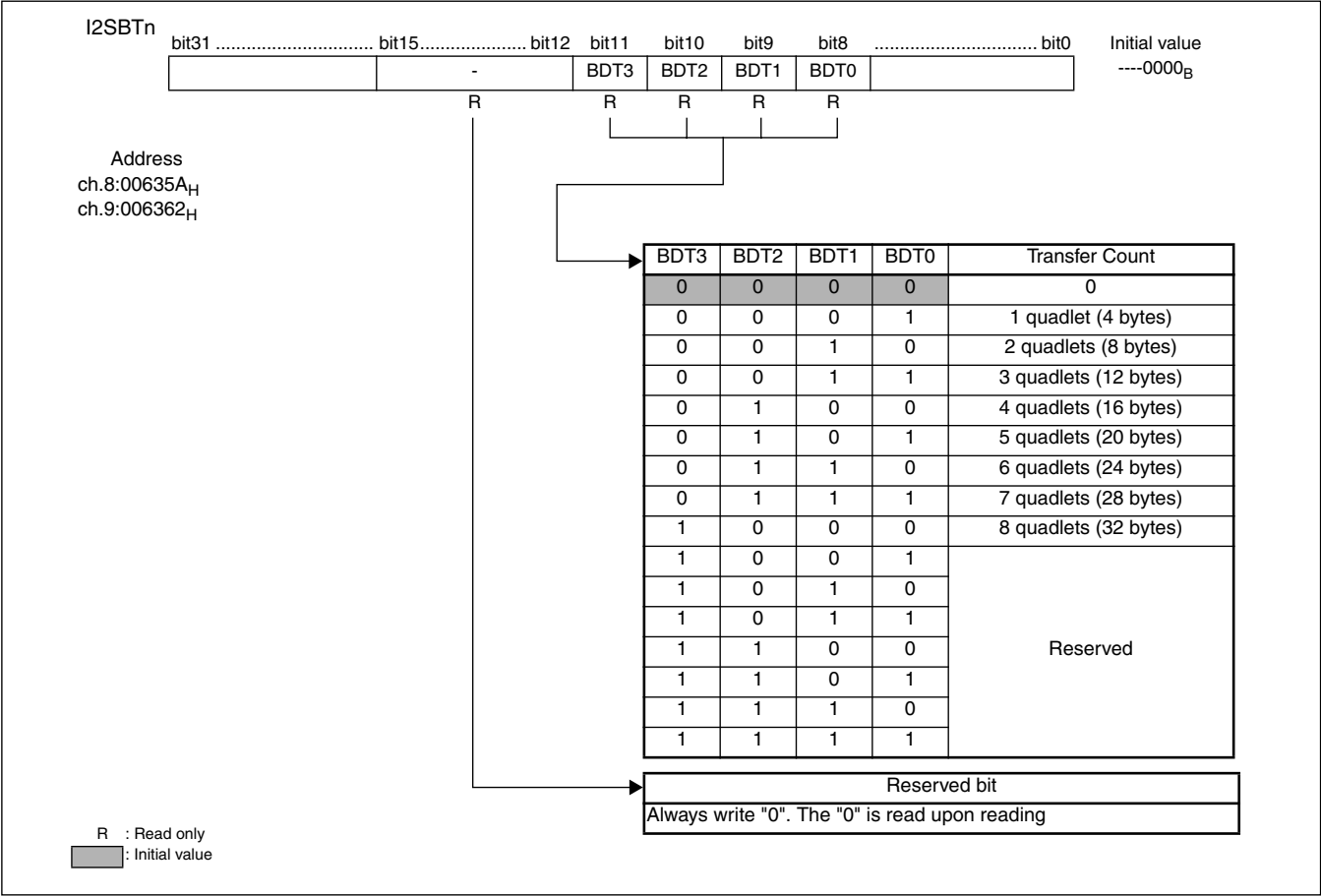


Table 61.2-6 Functional Descriptions on the I²S Buffer Transfer Count Register n (I2SBTn) Bits

Bit Name		Function
bit15-bit12	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit11-bit8	BDT: Transfer Count	<p>Indicates the number of valid data in the buffer. It is managed in 32-bit data units, and incremented by 1 when RTDTn:RD[7:0] is written and decremented by 1 when read. Writing "1" to I2SSCCR:SRST sets this bit to 0.</p> <p><Notes></p> <ul style="list-style-type: none"> • Data is handled per 32 bits as valid data. • Writing to RTDTn:RD[7:0] increments this bit by 1, regardless of I2SSCCR:D:DBL bit setting. • This bit is cleared to 0 if, while this bit is not 0, I2SSCRn:TR bit is changed. • Supported only in ch.8 and ch.9.

61.2.5 I²S Buffer Control Register n (I2SBCRn)

The I²S buffer control register n (I2SBCRn) enables the buffer and sets the transfer count for interrupt occurrences. Supported only in ch.8 and ch.9.

■ Bit configuration of the I²S buffer control register n (I2SBCRn)

Figure 61.2-5 shows the bit configuration of the I²S buffer control register n (I2SBCRn) and Table 61.2-7 shows the functions of the bits.

Figure 61.2-5 Bit Configuration of the I²S Buffer Control Register n (I2SBCRn)

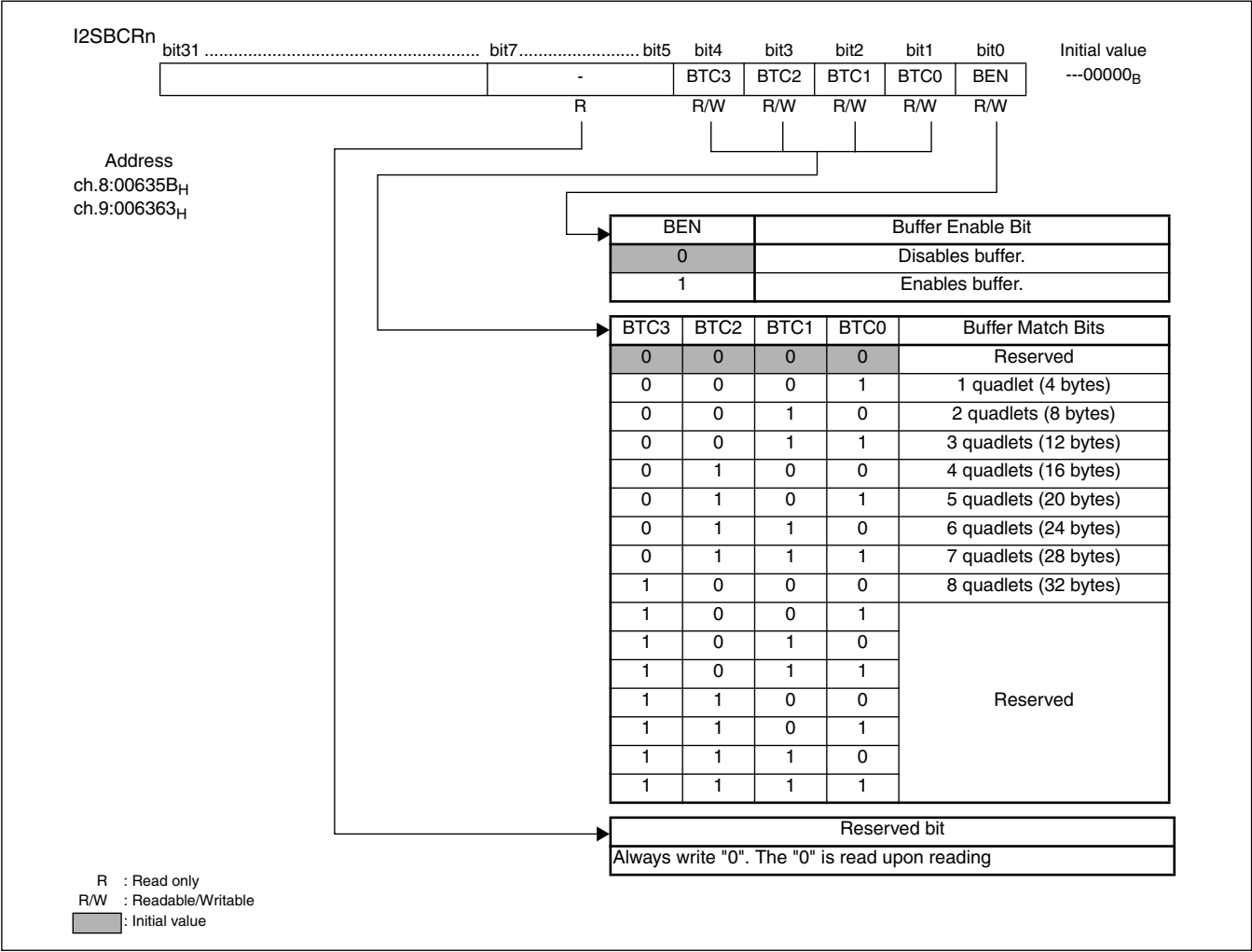


Table 61.2-7 Functional Descriptions on the I²S Buffer Control Register n (I2SBCRn) Bits

Bit Name		Function
bit7-bit5	Reserved bit	Reserved bit. The "0" is read upon reading. Write "0" when writing.
bit4-bit1	BTC: Buffer Match Bits	<p>When I2SBCRn:BEN="1", I2SBn:BDT[3:0] and this bit are compared, and I2SSCRn:RDYR is kept to "1" while either of the following conditions is met.</p> <ul style="list-style-type: none"> When I2SSCRn:TR="1" (transmission), I2SSCRn:RDYL is kept to "1" in the state where the number of data written to the left data buffer is smaller than the number of data set (I2SBn:BDT[3:0]I2SBCRn:BTC[3:0]) until LTDTn is written. When I2SSCRn:TR="1" (transmission), I2SSCRn:RDYR is kept to "1" in the state where the number of data written to the right data buffer is smaller than the number of data set (I2SBn:BDT[3:0]I2SBCRn:BTC[3:0]) until RTDTn is written. When I2SSCRn:TR="0" (reception), I2SSCRn:RDYL is kept to "1" in the state where I2SBn:BDT[3:0]=I2SBCRn:BTC[3:0] and LTDTn is written or the state where the number of data written to the left data buffer is greater than the number of data set (I2SBn:BDT[3:0]>I2SBCRn:BTC[3:0]). When I2SSCRn:TR="0" (reception), I2SSCRn:RDYR is kept to "1" in the state where I2SBn:BDT[3:0]=I2SBCRn:BTC[3:0] and RTDTn is written or the state where the number of data written to the right data buffer is greater than the number of data set (I2SBn:BDT[3:0]>I2SBCRn:BTC[3:0]). <p><Notes></p> <ul style="list-style-type: none"> When I2SSCRn:TR="0", setting this bit to 8 or above is prohibited. When I2SSCRn:TR="1", setting this bit to 9 or above is prohibited.
bit0	BEN: Buffer Enable Bit	<p>Buffer enable bit. Setting this bit to "1" enables the buffer. If this bit is set to "0" during buffer access, this bit becomes "0" after the buffer access.</p> <p><Notes></p> <ul style="list-style-type: none"> If this bit is set to "0" while valid data is being stored in the buffer, no data can be read from or written to the buffer. Setting this bit to "0" clears I2SBn:BDT to 0. Supported only in ch.8 and ch.9.

61.2.6 Left Data Register n (LTDTn)

The left data register n (LTDTn) is the transmission/reception buffer for the data for left.

■ Bit configuration of the left data register n (LTDTn)

Figure 61.2-6 shows the bit configuration of the left data register (LTDTn) and Table 61.2-8 shows the functions of the bits.

Figure 61.2-6 Bit Configuration of the Left Data Register n (LTDn)

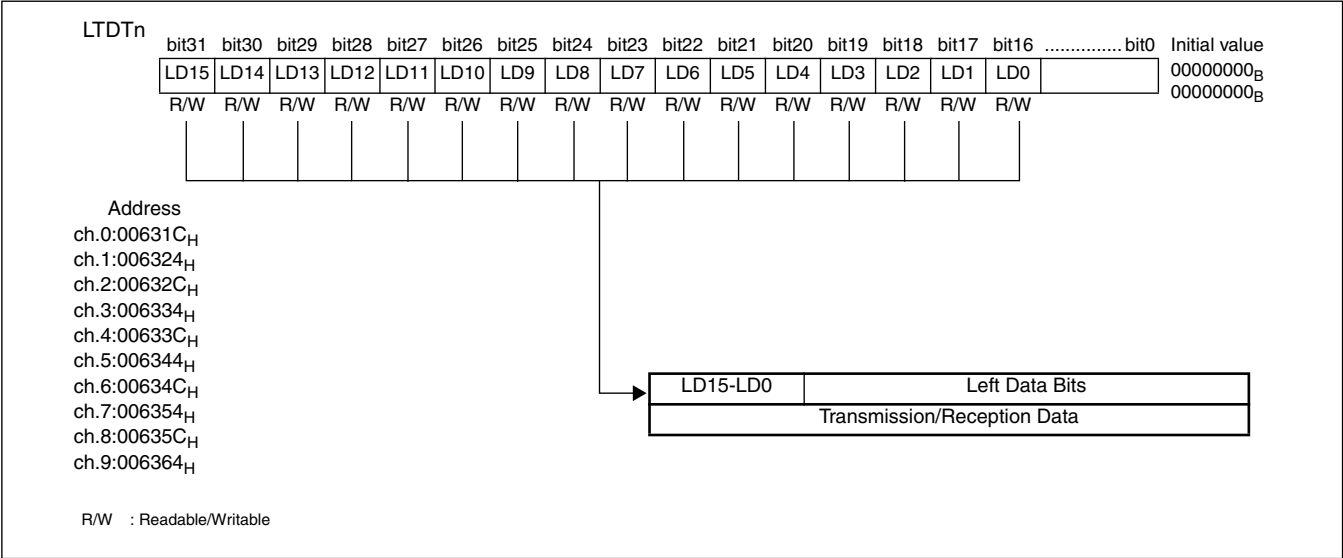


Table 61.2-8 Functional Descriptions on the Left Data Register n (LTDn) Bits

Bit Name		Function
bit31-bit16	LD15-LD0: Transmission/ Reception Data	<p>Stores transmission/reception data.</p> <p>In reception, the data received when WS pin is at "L" is stored. Data is stored from LD15 according to I2SSCCR:BL[3:0]. If data is less than 16 bits, a "0" is stored for an unreceived remaining bit.</p> <p>In transmission, the data to be output when WS pin is at "L" is written. Data is output from LD15 to SDn pin.</p> <p><Notes></p> <ul style="list-style-type: none">• In master mode, the I²S channels 0 to 7 cannot be read/written. Writing from software is disabled. 0x0000 is output upon reading.• In master mode, the I²S channels 8 and 9 cannot be read.• In slave mode, the I²S channels 0 to 7 cannot be read/written from the MediaLB. (Setting DMA mode in the MediaLB is prohibited.)• When I2SSCRn:MONO=0 and if writing transmission data, make sure to consecutively write 16-bit data to the left data register and right data register in this order.

61.2.7 Right Data Register n (RTDTn)

The right data register n (RTDTn) is the transmission/reception buffer for the data for right.

■ Bit configuration of the right data register n (RTDTn)

Figure 61.2-7 shows the bit configuration of the right data register (RTDTn) and Table 61.2-9 shows the functions of the bits.

Figure 61.2-7 Bit Configuration of the Right Data Register n (RTDTn)

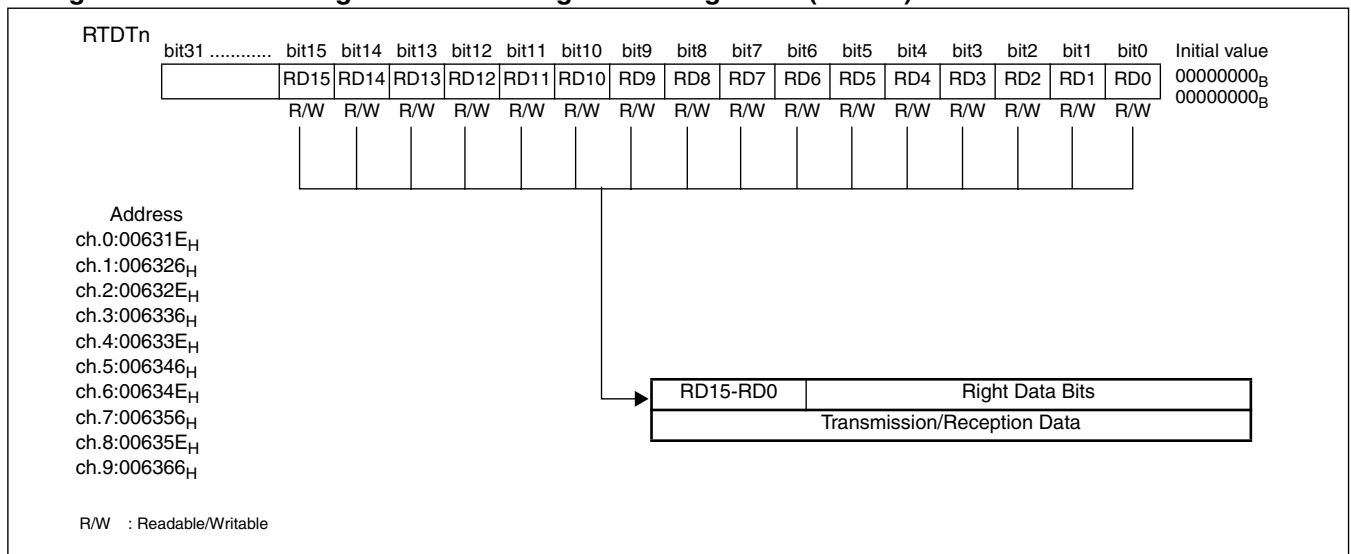


Table 61.2-9 Functional Descriptions on the Right Data Register n (RTDTn) Bits

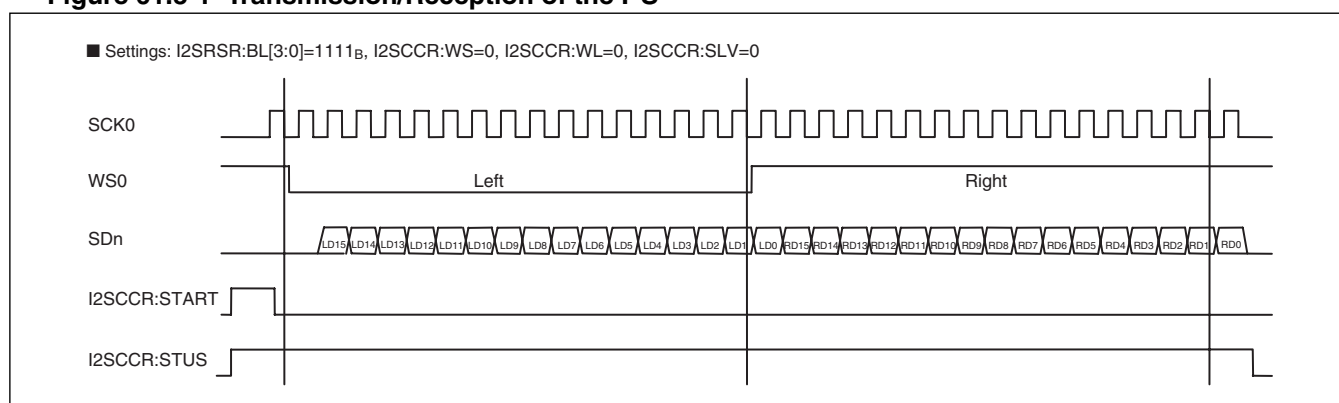
Bit Name		Function
bit15-bit0	RD15-RD0: Transmission/ Reception Data	<p>Stores transmission/reception data.</p> <p>In reception, the data received when WS pin is at "H" is stored. Data is stored from RD15 according to I2SCCR:BL[3:0]. If data is less than 16 bits, a "0" is stored for an unreceived remaining bit.</p> <p>In transmission, the data to be output when WS pin is at "H" is written. Data is output from RD15 to SDn pin.</p> <p><Notes></p> <ul style="list-style-type: none"> In master mode, the I²S channels 0 to 7 cannot be read/written. Writing from software is disabled. 0000_H is output upon reading. In master mode, the I²S channels 8 and 9 cannot be read. In slave mode, the I²S channels 0 to 7 cannot be read/written from the MediaLB. (Setting DMA mode in the MediaLB is prohibited.)

61.3. Operations of the I²S

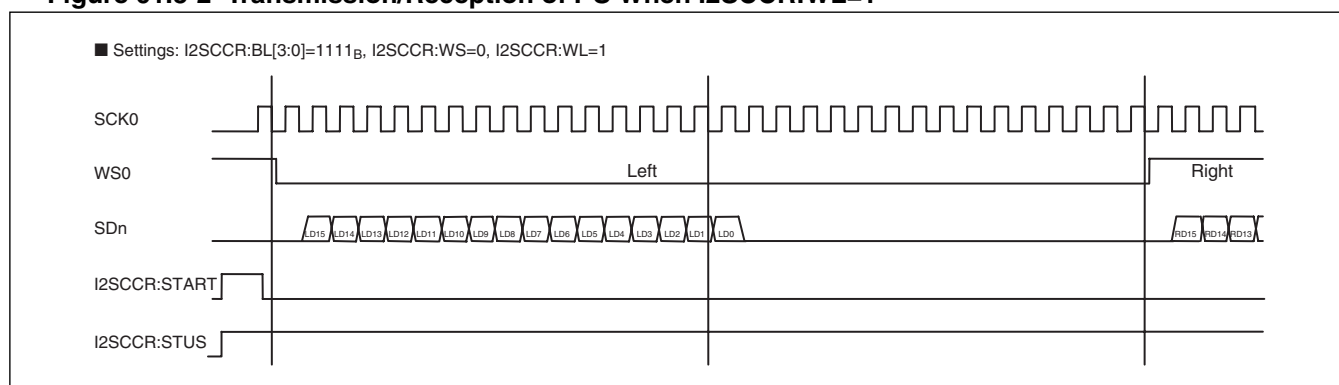
The I²S supports master mode/slave mode and transmits/receives data with MSB first.

■ Operations of the I²S

(1) Examples of I²S transmission/reception (master mode)

Figure 61.3-1 Transmission/Reception of the I²S

I²S (stands for Inter-Integrated Circuit Sound) sets WS pin to "L" at the falling edge of ISCK pin (or sets WS pin to "H" if I2SCCR:WS=1) and outputs bit 15 (LD15) of the data in the left register to SDn pin from the next falling edge of ISCK pin. The reception side samples SDn pin at the rising edge of ISCK pin and stores the data in the shift register. When a prescribed clock (specified in I2SRSR:BL[3:0], I2SCCR:WL) is generated at ISCK pin, WS pin is inverted at the falling edge of ISCK pin and bit 15 (RD15) of the data in the right register is output to SDn pin from the next falling edge of ISCK pin.

Figure 61.3-2 Transmission/Reception of I²S When I2SCCR:WL=1

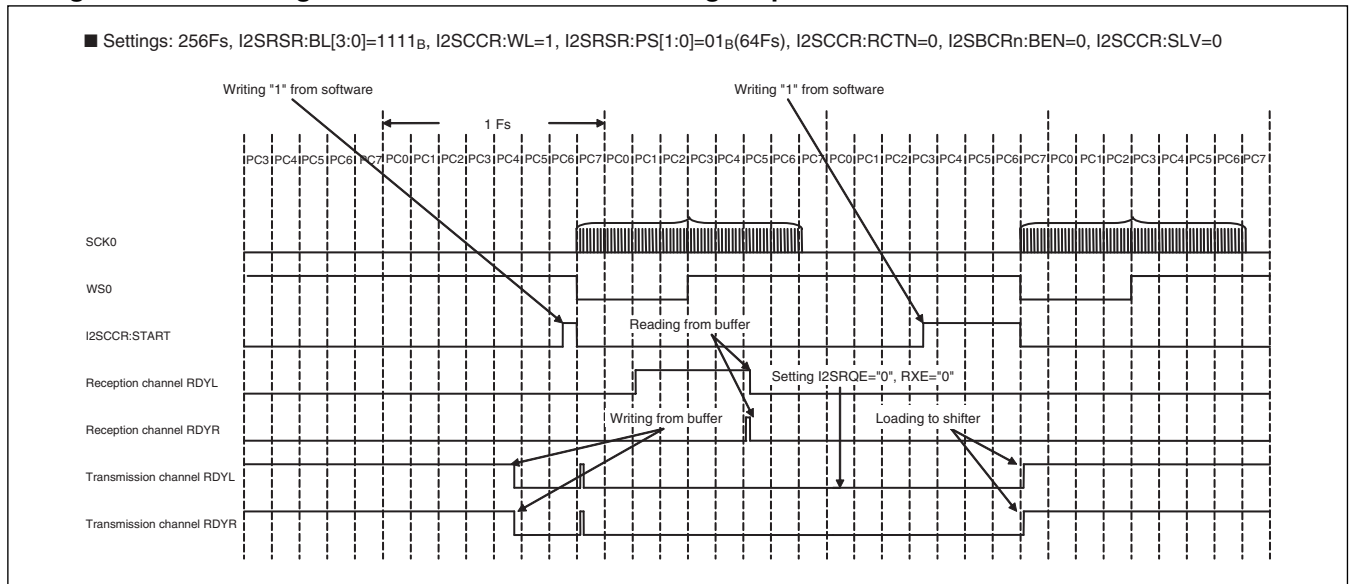
Setting to I2SCCR:WL=1 inverts WS pin at the twice the length set in I2SRSR:BL[3:0]. In transmission, an "L" is output for the remaining data after data of the bit length specified in I2SRSR:BL[3:0] is output. In reception, bits are received for the number of bits specified in I2SRSR:BL[3:0] and the remaining data will be discarded.

(2) Operations of the I²S with the MediaLB

A transmission/reception of the I²S starts when "1" is written to I2SCCR. START from software or "1" is written to I2SCCR:START from the buffer in order to generate a serial clock from ISCK pin. The timings of the MediaLB and I²S are shown below.

■ Single timing (I2CBCRn:BEN=0)

Figure 61.3-3 Timings of the MediaLB and I²S in Single Operation



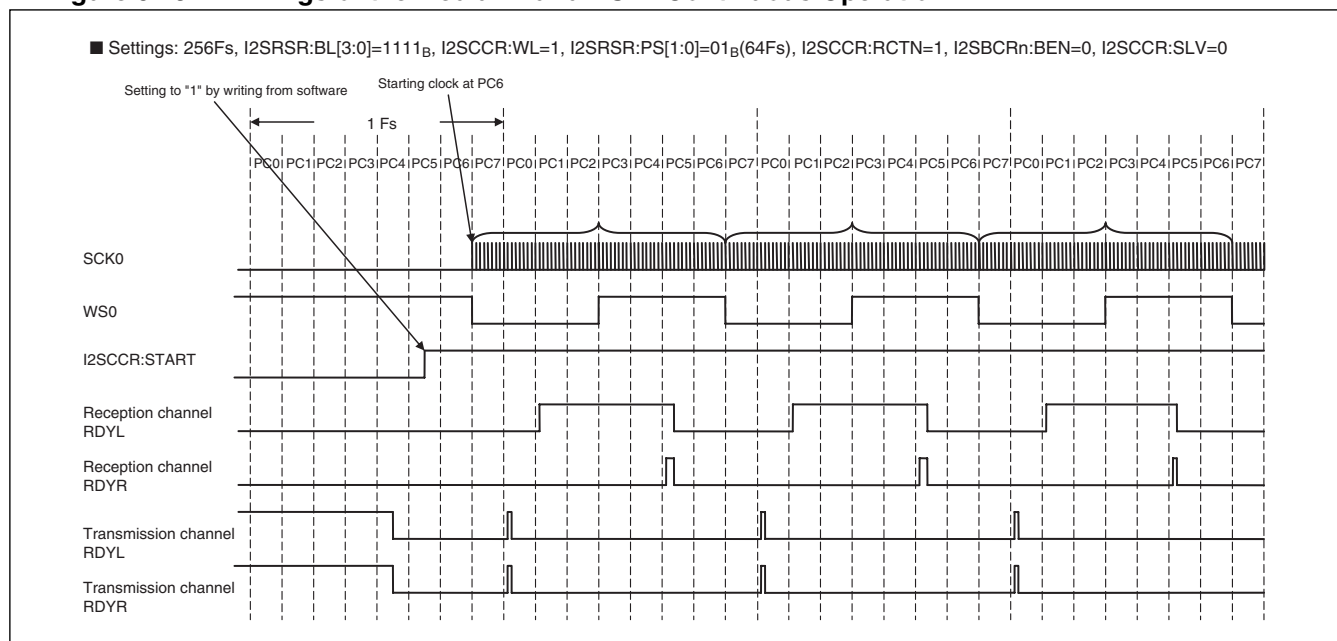
Setting the buffer enable bit I2SRQE to "1" writes data from the buffer to the right and left registers in the I²S, if RDYL and RDYR bits of the transmission channels are "1". Setting the I2SCCR:START bit to "1" sets WS pin to "L" simultaneously with the start of clock output to ISCK pin at PC7 and resets I2SCCR:START to "0". When transmission data is transferred to the transmission shifter, RDYL and RDYR bits of the transmission channels are set to "1", the transmission data is requested from the buffer, and then the transmission data is written from the buffer. If no valid transmission data exists in the buffer, 0000_H is written. When 2 bytes are received, RDYL of the transmission channel is set to "1". When 2 bytes are received again, RDYR of the transmission channel is set to "1", and the buffer reads the transmission data from I²S and writes it to the buffer.

<Notes>

- If the MediaLB consecutively receives data within one frame from the same channel, I²S transmission cannot be made in time and so a buffer overflow will occur in the local buffer of the MediaLB and the channel buffer of the buffer.
- In master mode, the I²S generates I²S serial clock (ISCK) from MLBCLK clock of the MediaLB. If the clock of MLBCLK is not supplied during I²S transmission/reception, I²S ISCK will stop in the middle of the transmission/reception.
- In master mode, the I²S performs a transmission/reception operation only when the MediaLB locks. (A serial clock is not generated at ISCK pin.)
- In master mode, when the MediaLB detects unlocking during transmission/reception, the I²S outputs clocks for the length of WS, and the serial clock stops.

■ Continuous timing (I2CBCRn:BEN=0)

Figure 61.3-4 Timings of the MediaLB and I²S in Continuous Operation



<Notes>

- If the MediaLB consecutively receives data within one frame from the same channel, I²S transmission cannot be made in time and so a buffer overflow will occur in the local buffer of the MediaLB and the channel buffer of the buffer.
- In master mode, the I²S generates I²S serial clock (ISCK) from MLBCLK clock of the MediaLB. If the clock of MLBCLK is not supplied during I²S transmission/reception, I²S ISCK will stop in the middle of the transmission/reception.
- In master mode, the I²S performs a transmission/reception operation only when the MediaLB locks. (A serial clock is not generated at ISCK pin.)
- In master mode, when the MediaLB detects unlocking during transmission/reception, the I²S outputs clocks for the length of WS, and the serial clock stops.

(3) Slave operation

If slave mode is set, ISCK pin and WS pin become input, and the I²S transmits/receives serial data according to the ISCK clock. In slave mode, accessing to the right/left registers from the MediaLB is disabled. Access is always made from software.

In transmission, when WS pin changes, data in the right/left registers is transferred to the shift register, and serial data is output in synchronization with the falling edge of ISCK pin. The data for the length specified in I2SRSR:WL is output, and if WS pin does not change, "L" is output to SDn pin. If WS pin changes before I2SRSR:WL specification, the data in the right/left registers is transferred to the shift register according to WS pin specification (WS pin is given priority).

In reception, serial data is taken into the shifter at the rising edge of ISCK pin, and the data is transferred to either right or left register from the shifter according to WS pin specification when the data for the length specified in I2SRSR:WL is received. If WS pin changes before I2SRSR:WL specification, the change of WS pin is detected and the data is transferred to either right or left register from the shifter.

61.4. Interrupts of the I²S

The I²S supports the interrupts for transmission/reception and error interrupts for error occurrences.

■ Interrupt sources of the I²S

Interrupt flags and interrupt sources are shown below.

Transmission /Reception	Interrupt Flag	Register	Interrupt Source	Interrupt Enable Bit	Interrupt Flag Clear
Reception	RDYL/RDYR	I2SSCRn	Writing reception data to register	I2SSCRn:IEL/IER	Reading transmission data
	ERR		Overrun error	I2SSCRn:IEE	Writing "0" to I2SSCRn:ERR bit
Transmission	RDYL/RDYR	I2SSCRn	Transmission data empty	I2SSCRn:IEL/IER	Writing transmission data
	ERR		Underrun error	I2SSCRn:ERR	Writing "0" to I2SSCRn:ERR bit

61.4.1 Reception Interrupts

In reception mode, each flag of I2SSCRn is set to "1" when the following operation is performed.

- Data reception complete: When I2SBCRn="0", writing reception data from the reception shift register to the left data register sets RDYL flag to "1".
When I2SBCRn="0", writing reception data from the reception shift register to the right data register sets RDYR flag to "1".
When I2SBCRn="1", writing reception data from the reception shift register to the left data register by setting I2SBTn:BDT=I2SBTn:BTC sets RDYL flag to "1".
When I2SBCRn="1", writing reception data from the reception shift register to the right data register by setting I2SBTn:BDT=I2SBTn:BTC sets RDYR flag to "1".
- Overrun error:When I2SBCRn="0", writing reception data to the left data register by setting RDYL="1" sets ERR flag to "1".
When I2SBCRn="0", writing reception data to the right data register by setting RDYR="1" sets ERR flag to "1".
When I2SBCRn="1", if writing of reception data occurs in a buffer full state, ERR flag is set to "1". The reception data in this state cannot be written and so will be discarded.

An interrupt occurs under the following conditions:

- I2SSCRn:RDYL="1" when I2SSCRn:LEL="1"
- I2SSCRn:RDYR="1" when I2SSCRn:IER="1"
- I2SSCRn:ERR="1" when I2SSCRn:IEE="1"

RDYL/RDYR flags are cleared to "0" upon reading from the right/left data registers. ERR flag is cleared to "0" when writing "0" to the ERR flag.

61.4.2 Transmission Interrupts

In transmission mode, each flag of I2SSCRn is set to "1" when the following operation is performed.

- Transmission data empty:When transmission data exists in the left data register, transferring the transmission data from the left data register to the transmission shift register sets RDYL flag to "1".
When transmission data exists in the right data register, transferring the transmission data from the right data register to the transmission shift register sets RDYR flag to "1".

-Underrun error:When data is written to the right/left data registers by setting I2SSCR:RCTN="1" and then transmission shift register becomes empty and requests transmission data from the right/left data registers while RDYL/RDYR flags are "1", ERR flag is set to "1".

An interrupt occurs under the following conditions:

- I2SSCRn:RDYL="1" when I2SSCRn:LEL="1"
- I2SSCRn:RDYR="1" when I2SSCRn:IER="1"
- I2SSCRn:ERR="1" when I2SSCRn:IEE="1"

RDYL/RDYR flags are cleared to "0" when writing data to the right/left data registers. ERR flag is cleared to "0" when writing "0" to the ERR flag.

61.5. I²S Buffer

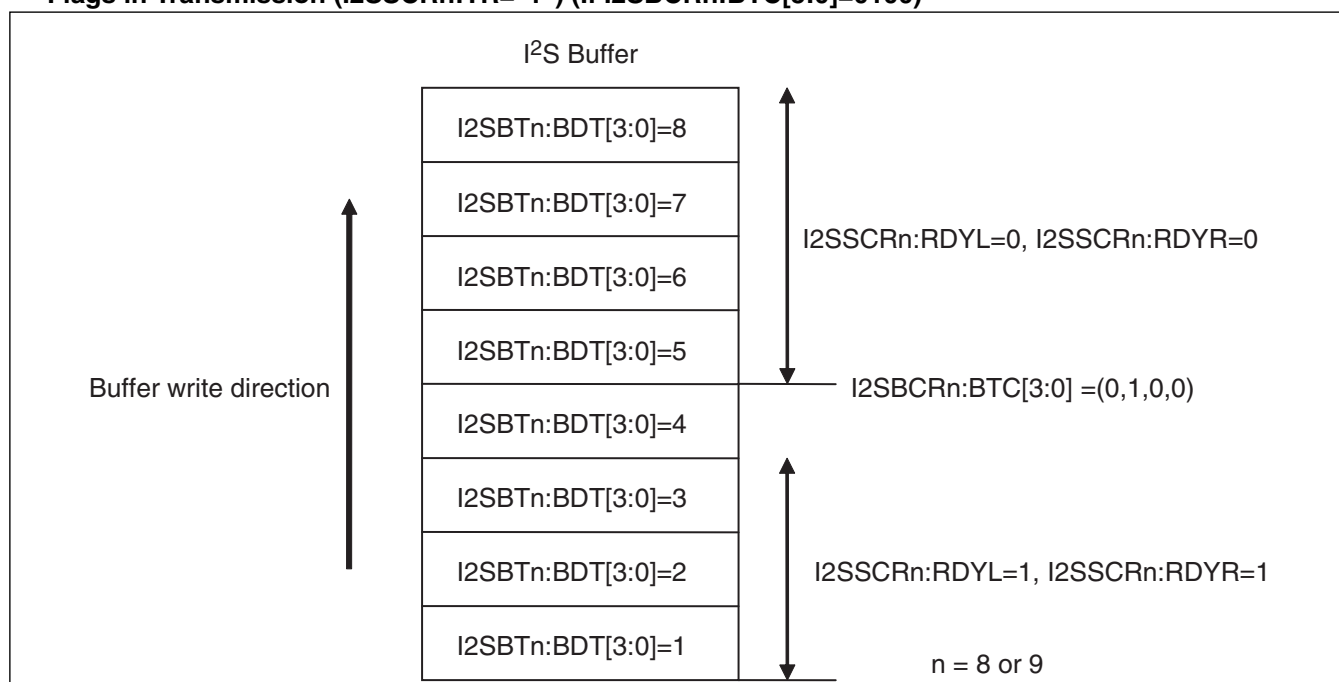
The I²S has eight buffers for ch.8 and ch.9 and can set the timing of interrupt occurrences by the I²S buffer control register n.

■ Relationship between BTC3 to BTC0 in the I²S Buffer Control Register n (I2SBCRn) and interrupts

<I2SSCRn:TR="1"(transmission)>

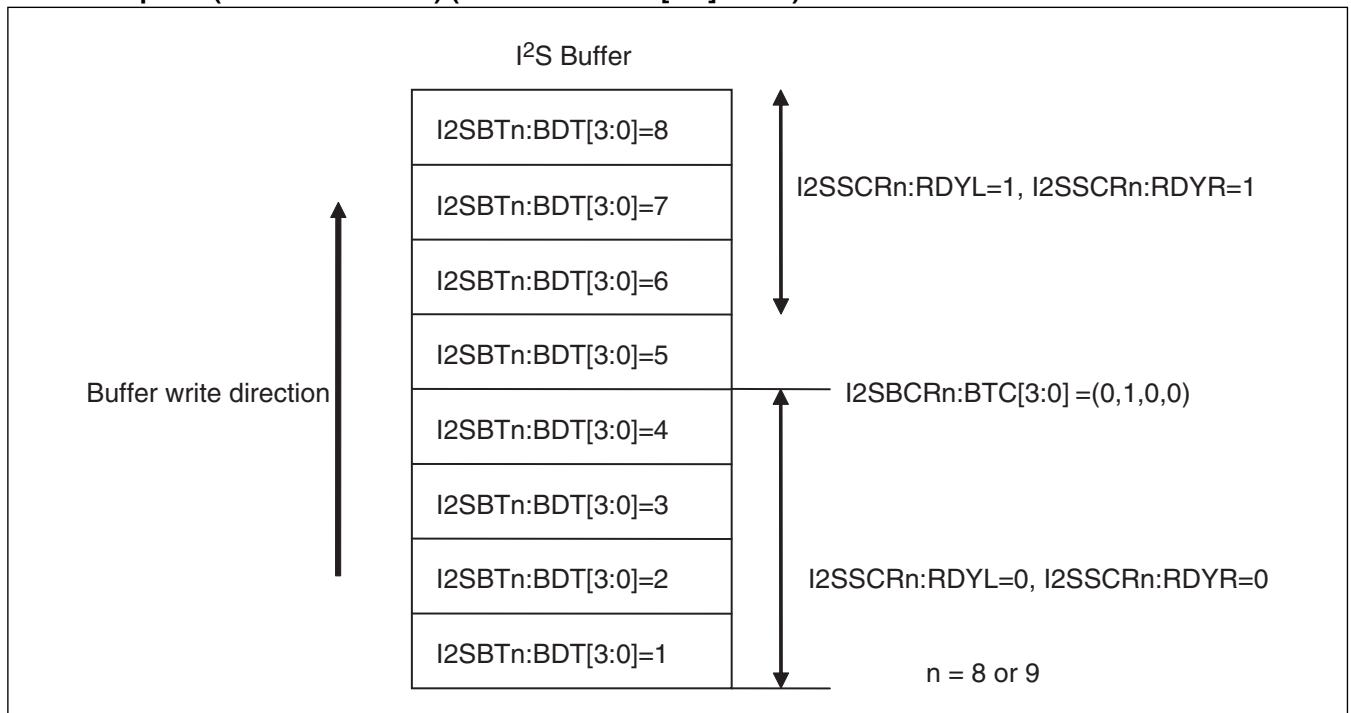
<I2SSCRn:TR="0"(reception)>

Figure 61.5-1 Relationship between the BTC Bits in the I²S Buffer Control Register and the Interrupt Flags in Transmission (I2SSCRn:TR="1") (If I2SBCRn:BTC[3:0]=0100)



In transmission (I2SSCRn:TR="1"), if the number of data written to the data buffer is smaller than the number of data set (I2SBTn:BDT[3:0]<I2SBCRn:BTC[3:0]), interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "1". The above figure explains the case where I2SBCRn:BTC[3:0] is (0100). Writing transmission data to the right/left registers increments I2SBTn:BDT by 1. If I2SBTn:BDT[3:0] becomes 4 or greater, interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "0". Transferring transmission data in the I²S data buffer to the I²S transmission shift register decrements I2SBTn:BDT by 1. If it becomes 3 or less, interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "1".

Figure 61.5-2 Relationship between the BTC Bits in I²S Buffer Control Register and the Interrupt Flags in Reception (I2SSCRn:TR="0") (If I2SBCRn:BTC[3:0]=0100)



In reception (I2SSCRn:TR="0"), if the number of data written to the data buffer is greater than the number of data set (I2SBTn:BDT[3:0]<I2SBCRn:BTC[3:0]), interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "1". The above figure explains the case where I2SBCRn:BTC[3:0] is (0100). Receiving data from the I²S writes the reception data to the I²S buffer and increments I2SBTn:BDT by 1. If I2SBTn:BDT[3:0] then becomes 5 or greater, interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "1". Reading reception data in the I²S data buffer from software decrements I2SBTn:BDT by 1. If I2SBTn:BDT[3:0] then becomes 4 or less, interrupt flags (I2SSCRn:RDYL, I2SSCRn:RDYR) are set to "0".

<Notes>

- In transmission (I2SSCRn:TR="1"), setting I2SBCRn:DTC (n is 8 or 9) to 9 or larger is prohibited.
- In reception (I2SSCRn:TR="0"), setting I2SBCRn:DTC (n is 8 or 9) to 8 or larger is prohibited.

Chapter 62 USB Mini-Host

This chapter describes the functions and operations of USB Mini-host.

62.1. USB Mini-host features

USB Mini-host provides minimum host functionalities required. It enables data to be transferred to and from a device without PC intervention.

■ USB Mini-host features

USB Mini-host has the following features:

- Automatic detection of Full Speed / Low Speed devices.
- Support Full Speed transfer.
- Automatic detection of device connection and disconnection.
- Support of reset sending function to USB bus.
- IN/OUT/SETUP/SOF token support.
- Automatic transmission of handshake packet for IN token (excluding STALL).
- Handshake packet automatic detection at OUT token.
- Support a maximum data packet length of 256 bytes.
- Supports various error handling (CRC error/toggle error/time-out).
- Wake-up function support.

62.2. Differences in USB Mini-host

Description of the differences between the standard USB host and USB Mini-host.

■ USB Host versus USB Mini-host

		Host	Mini-host
Support Hub		○	×
Transfer	Bulk transfer	○	○
	Control transfer	○	○
	Interrupt transfer	○	○
	Isochronous transfer	○	×
Transfer speed	Low Speed	○	×
	Full Speed	○	○
PRE packet support		○	×
SOF packet support		○	○
Error	CRC error	○	○
	Toggle error	○	○
	Time-out	○	○
	Max. packet < Receive Data	○	○
Detection of connection and disconnection of devices		○	○
Transfer speed detection		○	○

○: Supported

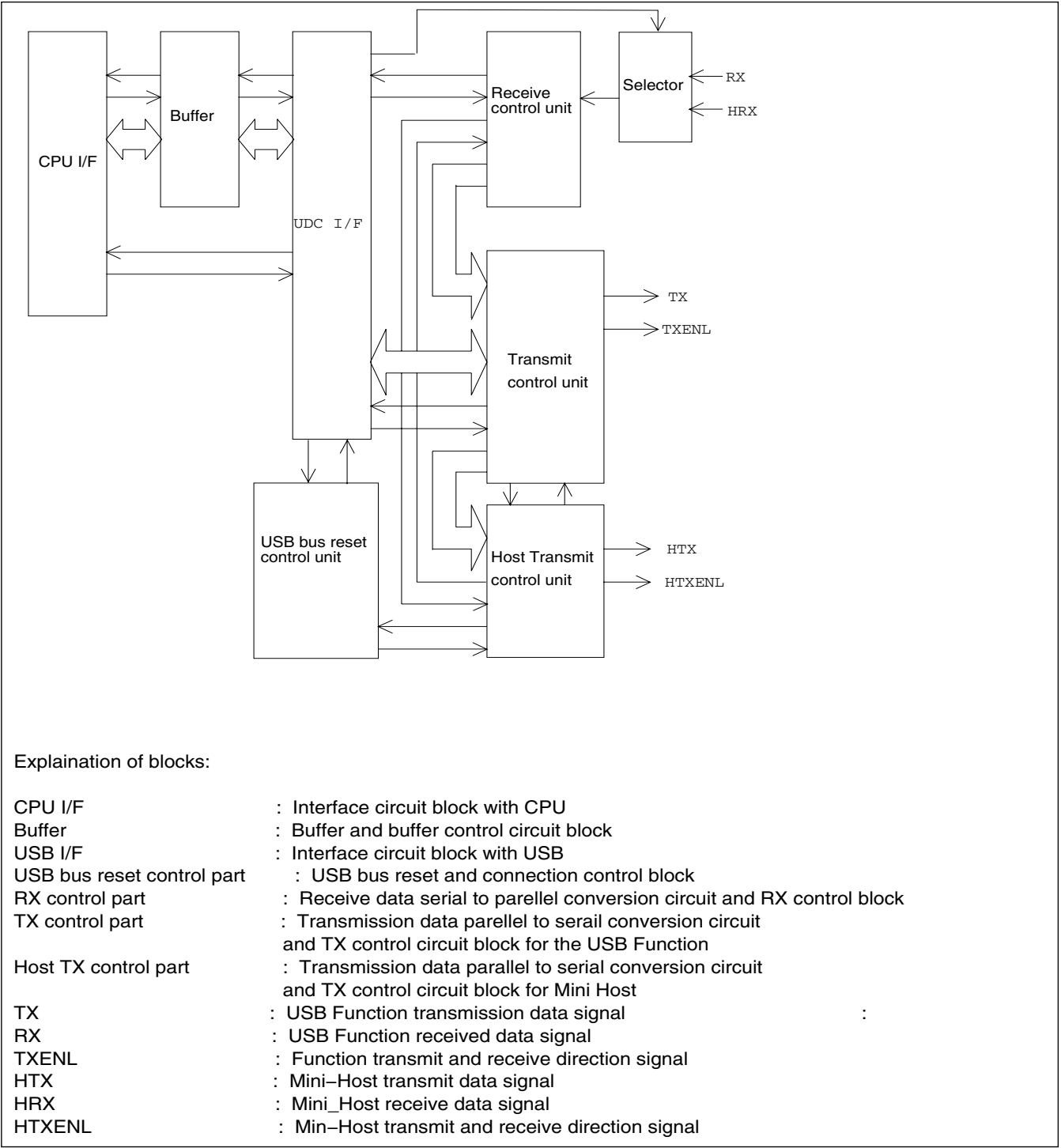
×: Not supported

62.3. USB Mini-host Block Diagram

Block diagram of USB Mini-host.

■ Block Diagram of USB Mini-host

Figure 62.3-1 Block Diagram of USB Mini-host



62.4. USB Mini-host Registers

This chapter describes all the registers of the USB Mini-host function.

■ Register of USB Mini-host

Table 62.4-1 Registers of USB Mini-host

bit	7	6	5	4	3	2	1	0	
	RWKIRE	URIRE	CMPIRE	CNNIRE	DIRE	SOFIRE	URST	HOST	HCNTLn
bit	15	14	13	12	11	10	9	8	
	Reserved					SOFSTEP	CANCEL	RETRY	HCNTHn
bit	7	6	5	4	3	2	1	0	
	TCAN	Reserved	RWKIRQ	URIRQ	CMPIRQ	CNNIRQ	DIRQ	SOFIRQ	HIRQn
bit	15	14	13	12	11	10	9	8	
	LSTSOE	RERR	TOUT	CRC	TGERR	STUFF	HS		HERRn
bit	7	6	5	4	3	2	1	0	
	Reserved					SOFBUSY	SUSP	TMODE	HSTATEn
bit	15	14	13	12	11	10	9	8	
	FCMP								HFCOMPn
bit	7	6	5	4	3	2	1	0	
	RTIM0								HRTIMER0n
bit	15	14	13	12	11	10	9	8	
	RTIM1								HRTIMER1n
bit	7	6	5	4	3	2	1	0	
	Reserved						RTIM2		HRTIMER2n
bit	15	14	13	12	11	10	9	8	
	Reserved	HADR							HADRn
bit	7	6	5	4	3	2	1	0	
	EOF0								HEOFLn
bit	15	14	13	12	11	10	9	8	
	Reserved		EOF1						HEOFHn
bit	7	6	5	4	3	2	1	0	
	FRAME0								HFRAMELn
bit	15	14	13	12	11	10	9	8	
	Reserved					FRAME1			HFRAMEHn
bit	7	6	5	4	3	2	1	0	
	TGGL	TKEN			EDPT				HTOKENn

62.4.1 Host Control Register (HCNTn)

Host control register (HCNTn) specifies the USB operation mode and the interrupt settings.

■ Host Control Register Low (HCNTLn)

Figure 62.4-1 Host Control Register Low (HCNTLn)

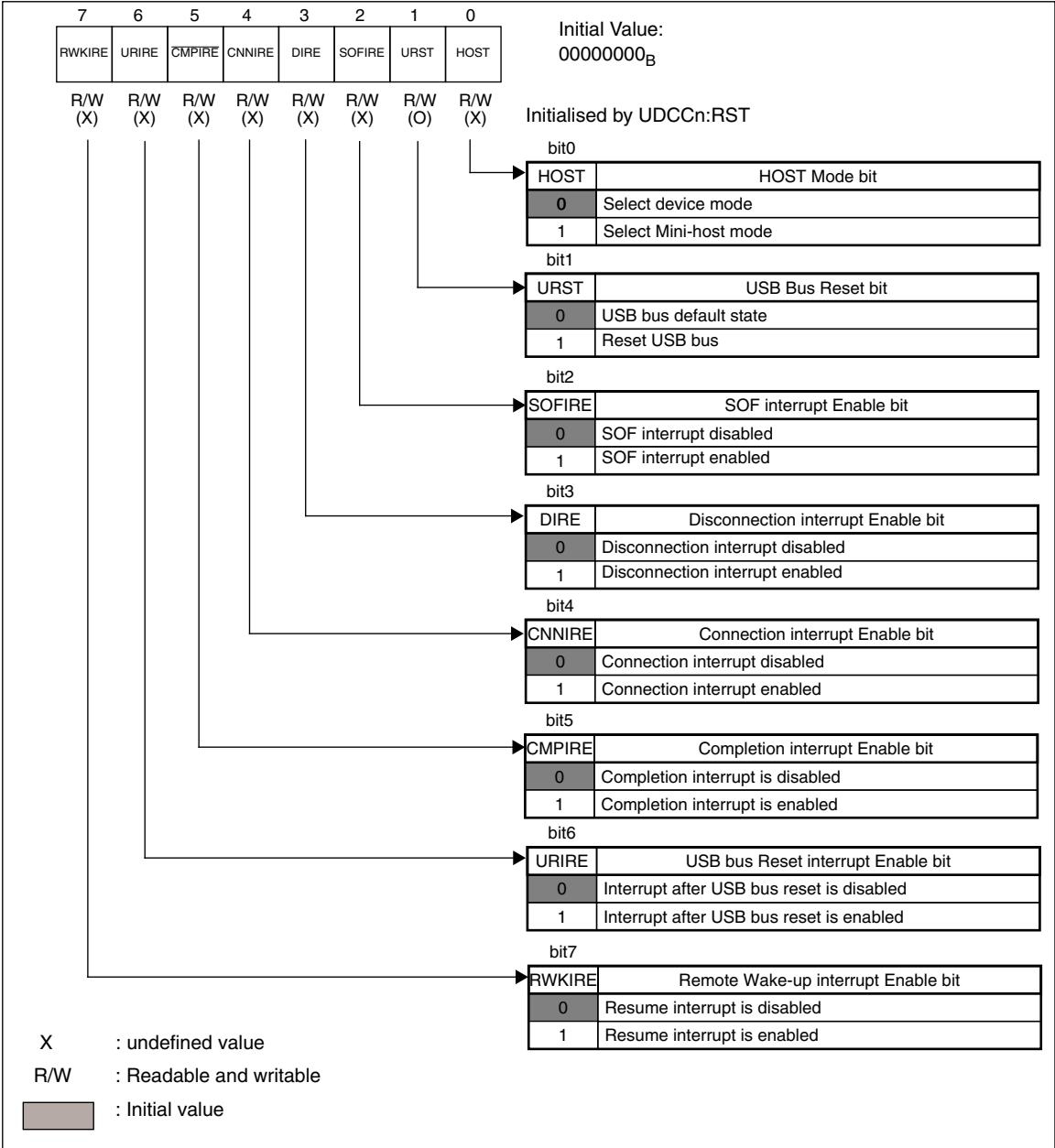


Figure 62.4-2 Host Control Register High (HCNTnH)

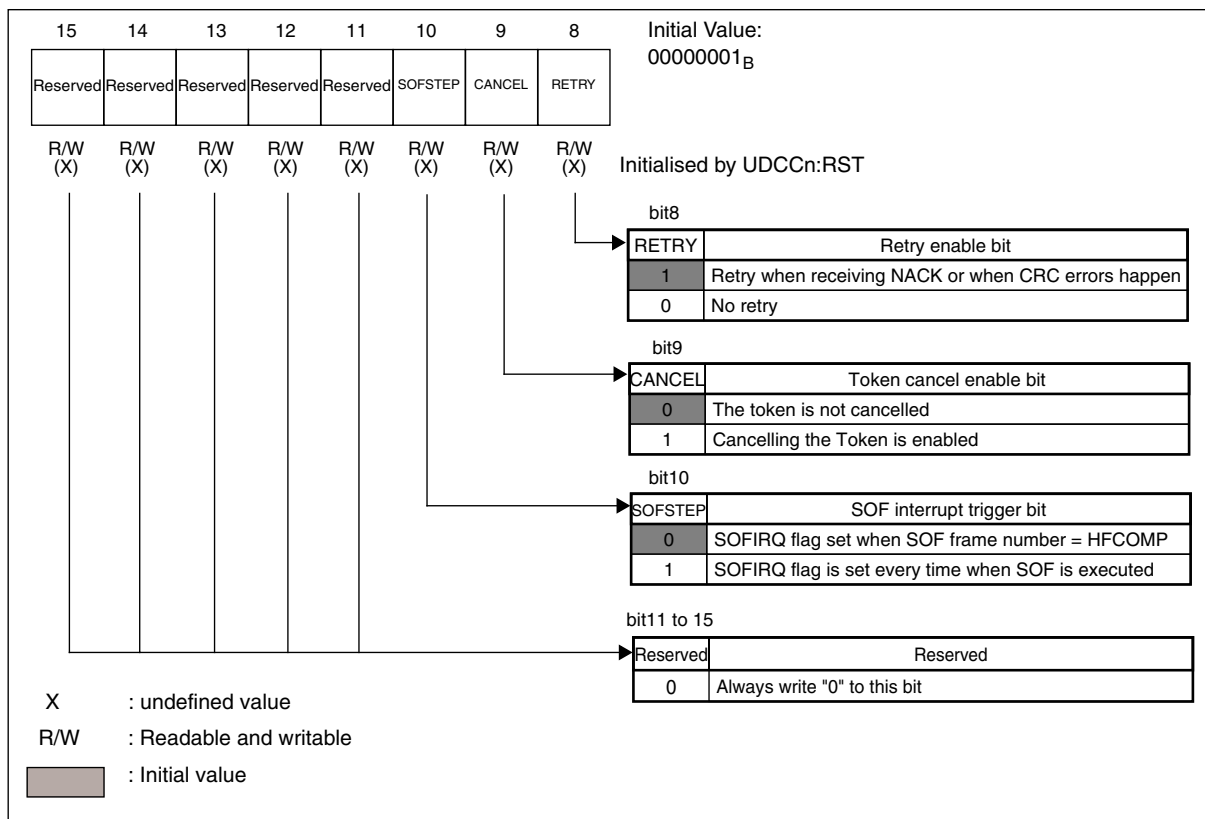


Table 62.4-2 Functional Description of each bit of the HCNTn registers:

	Bit names	Function
Bit15-11	Reserved	Please write "0" to these bits
Bit10	SOFSTEP	<p>This bit sets whether an interrupt due to SOF is generated every time SOF is executed (when set to "1").</p> <p>The interrupt is enabled if HCNTLn:SOFIRE bit is set.</p> <p>If SOFSTEP is "0", SOF Interrupt request flag HIRQn:SOFIRQ is set when the lower 8 bits of the SOF Frame number are equal to the value set in the SOF Interruption Comparison Register (HFCOMPn). If SOFSTEP is "1", SOF Interruption Request Flag HIRQn:SOFIRQ is set whenever SOF is executed. But HIRQn:SOFIRQ bit is not set by the SOF token executed by setting HTOKENn:TKEN bits to "001B". This bit is not initialized with the RST bit in the UDC control register (UDCCn).</p>
Bit9	CANCEL	<p>This bit is used to cancel a token.</p> <p>If this bit is set to "1", when HTOKENn:TKEN bits is set to execute a token in EOF area (it is specified by the HEOFn register), the token is cancelled.</p> <p>If this bit is set to "0", the token indicated by HTOKENn:TKEN bits is not cancelled.</p> <p>To confirm if any token was cancelled, read HIRQn:TCAN bit.</p> <p>UDCCn:RST does not initialize this bit.</p>

MB91460 Series

	Bit names	Function
Bit8	RETRY	<p>This bit is used to enable the retry of the token.</p> <p>If this bit is set to "1", when a NAK is received from the usb-device or various errors occur (HERRn:RERR="1", HERRn:TOUT="1", HERRn:CRC="1", HERRn:TGERR="1", HERRn:STUFF="1"), USB mini-host retries the token.</p> <p>The retry is executed for the time period specified in Retry Timer Setting Registers (HRTIMERHn, HRTIMERMn, HRTIMERLn).</p> <p>This bit is not initialized with the RST bit in the UDC control register (UDCCn).</p>
Bit7	RWKIRE	<p>This bit is used to enable Remote Wake-up Interrupt.</p> <p>If this bit is set to "1", when the HIRQn:RWKIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:RWKIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit6	URIRE	<p>This bit is used to enable USB bus Reset Interrupt.</p> <p>If this bit is set to "1", when the HIRQn:URIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:URIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit5	CMPIRE	<p>This bit is used to enable Completion Interrupt Request.</p> <p>If this bit is set to "1", when the HIRQn:CMPIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:CMPIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit4	CNNIRE	<p>This bit is used to enable Connection Interrupt</p> <p>If this bit is set to "1", when the HIRQn:CNNIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:CNNIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit3	DIRE	<p>This bit is used to enable Disconnection Interrupt.</p> <p>If this bit is set to "1", when the HIRQn:DIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:DIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit2	SOFIRE	<p>This bit is used to enable SOF Interrupt.</p> <p>If this bit is set to "1", when the HIRQn:SOFIRQ bit is set to "1", the related interrupt is triggered</p> <p>If this bit is set to "0", no interrupt is generated when the HIRQn:SOFIRQ bit is set to "1", UDCCn:RST does not initialize this bit.</p>
Bit1	URST	<p>Writing "1" to this bit resets the USB bus. It is cleared when USB bus reset is completed.</p> <p>Reading "1" indicates that the USB bus is still being reset.</p> <p>Writing "1" to this bit during UDCCn:RST="1" is ignored.</p> <p>It is forbidden to set this bit when the HSTATEn:SUSP bit is set or while a token is being executed.</p> <p>It is also forbidden to update HCNTLn or HCNTHn while this bit is set.</p>
Bit0	HOST	<p>This bit controls the USB function to act as a Device or as a Mini-Host. Setting "1" to this bit selects the Mini-Host mode and setting "0" selects Function mode. Because it takes time to transit mode after updating this bit, please read this bit to confirm if the mode transition is completed.</p> <p>This bit must be changed when RST bit is set to "1".</p> <p>To change the function mode from device to host function, a disconnection of the HOST PC by setting the UDCCn:HCONX bit must be performed before updating the HOST bit.</p> <p>To change the function mode from Host mode to Function mode, clear the HSTATEn:SOFBUSY bit to "0", set the HTOKENn:TKEN bits to "000B", and clear the HSTATEn:SUSP bit to "0", before updating the HOST bit.</p> <p>UDCCn:RST does not initialize this bit.</p>

62.4.2 Host Interrupt Register (HIRQn)

The host interrupt register (HIRQn) displays all the interrupt request flags of USB mini-Host. The interrupt is enabled only when the corresponding interrupt enable bit in the host control register (HCNTLn/HCNTHn) is enabled except for the TCAN bit.

■ Host Interrupt Register (HIRQn)

Figure 62.4-3 Bit Configuration of Host Interrupt Register (HIRQn)

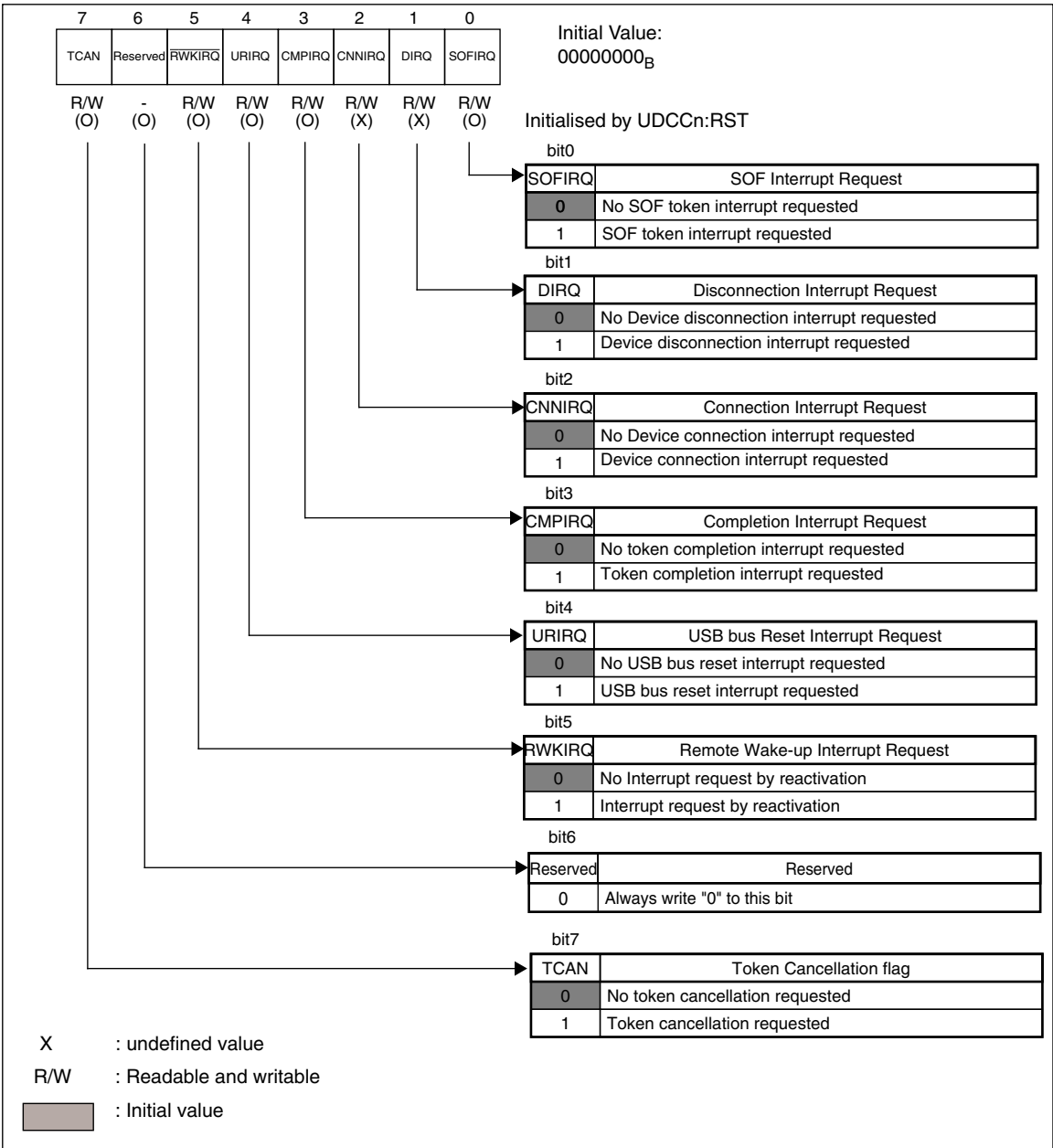


Table 62.4-3 Functional Description of each bit in the HIRQn registers:

	Bit names	Function
Bit7	TCAN	When a token is cancelled by the setting of HCNTLn:CANCEL bit, this bit is set to "1". If this bit is "0", it indicates that no token has been cancelled. This bit is cleared by writing "0". Writing "1" to this bit is ignored. No interrupt is raised when this bit is set to "1". If you want to handle this bit in an interrupt routine, please check this bit in the interrupt routine called by SOF interrupt. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initialized.
Bit6	Reserved	Please write "0" to this bit.
Bit5	RWKIRQ	This bit is set to "1" when the resume operation is completed, and an interrupt is triggered if HCNTLn:RWKIRE bit is "1". If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initialized.
Bit4	URIRQ	This bit is set to "1" when the reset operation of USB bus has been completed, and an interrupt is triggered if HCNTLn:URIRES bit is "1". If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initialized.
Bit3	CMPIRQ	This bit is set to "1" when a token has been completed, and an interrupt is triggered if HCNTLn:CMPIRES bit is "1". If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initialized. This flag is not set to "1" when the HIRQn:TCAN bit is set to "1".
Bit2	CNNIRQ	This bit is set to "1" when a device connection has been detected, and an interrupt is triggered if HCNTLn:CNNIRES bit is "1". If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. It is not initialized with the UDCCn:RST bit. This bit works also in the USB device mode, not only in the USB mini-host mode.
Bit1	DIRQ	This bit is set to "1" when the device disconnection has been detected, and an interrupt is triggered if HCNTLn:DIRE bit is "1". If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. It is not initialized with the UDCCn:RST bit. This bit works also in the USB device mode, not only in the USB mini-host mode.
Bit0	SOFIRQ	When a SOF transmission starts, in the case HCNTLn:SOFSTEP is "0", the FRAME Number in least significant 8 bits of the SOF token and HFCOMP are compared. And if a match is detected, this bit is set to "1". In the case the HCNTLn:SOFSTEP bit is "1", this bit is set to "1" every time a SOF transmission starts. When this bit is set to "1" and if the HCNTLn:SOFIRE bit is "1", an interrupt is generated. If this bit is "0", it doesn't indicate anything. This bit is cleared by writing "0". Writing "1" preserves the current state. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initialized.

62.4.3 Host Error Status Register (HERRn)

The host error status register (HERRn) indicates whether an error has occurred while sending or receiving data in host mode.

■ Host Error Status Register (HERRn)

Figure 62.4-4 Bit Description of the Host Error Status Register (HERRn)

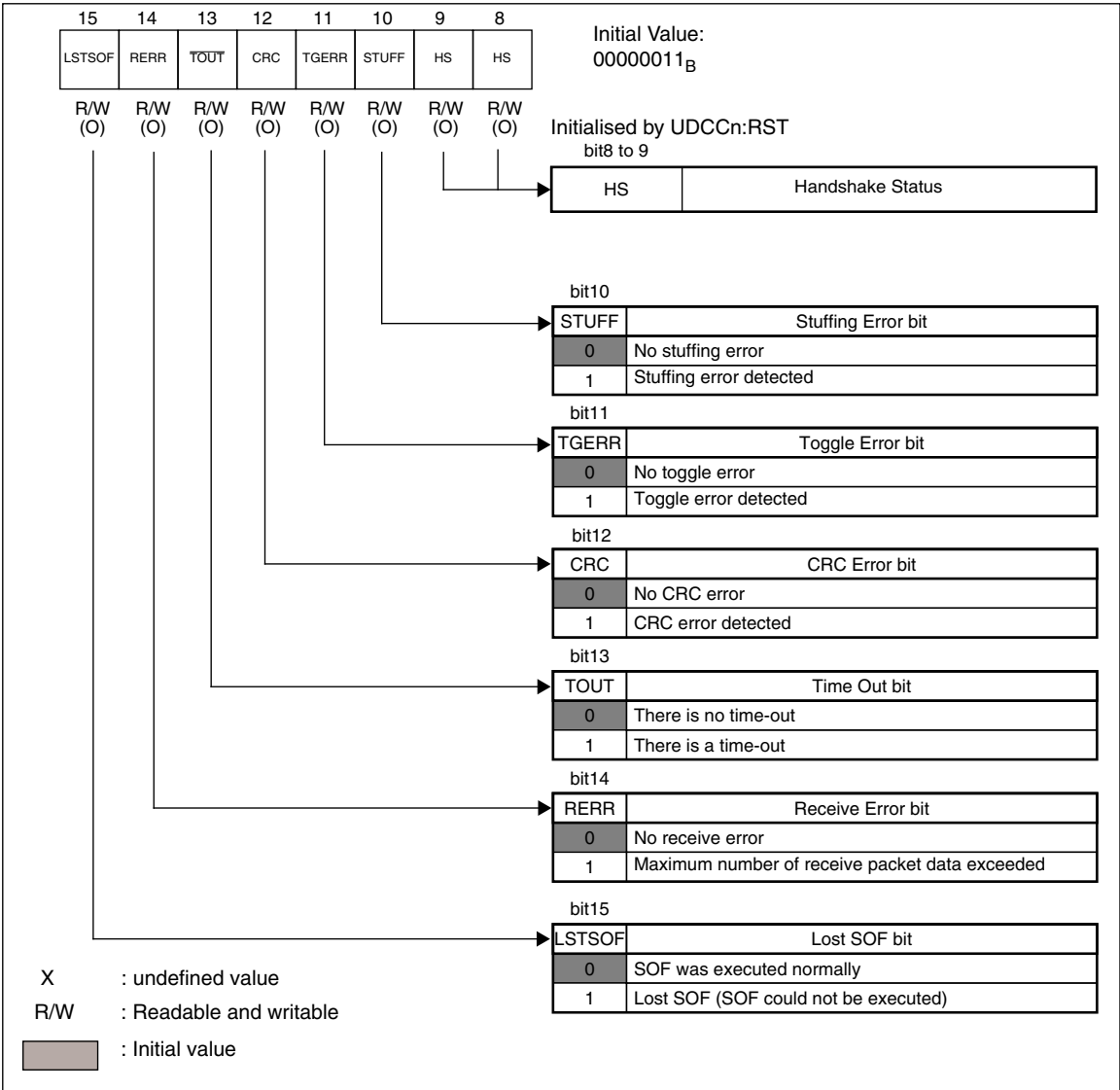


Table 62.4-4 Functional Description of each bit of the HERRn registers:

	Bit names	Function															
Bit15	LSTSOF	In the case the read value of this bit is "1", it indicates that a SOF token could not be executed because another token was running when trying to execute it in host mode. In the case the read value of this bit is "0", it indicates that no Lost SOF Error has been detected. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit14	RERR	In the case the read value of this bit is "1", it indicates that the number of packet received has exceeded a maximum Packet Size setting in host mode. When this bit is set to "1", the HERRn:TOUT bit is also set to "1" at the same time, In the case the read value of this bit is "0", it indicates that no such a error has occurred.. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit13	TOUT	In the case the read value of this bit is "1", it indicates that no response from the device has been received for a certain time, in Host mode. In the case the read value of this bit is "0", it indicates that no time out has detected. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit12	CRC	In the case the read value of this bit is "1", it indicates that a CRC error has occurred in Host mode. When this bit is set to "1", the HERRn:TOUT bit is also set to "1" at the same time, In the case the read value of this bit is "0", it indicates that no such a error has occurred.. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit11	TGERR	In the case the read value of this bit is "1", it indicates that the received toggle has not matched with the setting in Host mode. In the case the read value of this bit is "0", it indicates that such a error has occurred. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit10	STUFF	In the case the read value of this bit is "1", it indicates that a CRC error has occurred in Host mode. When this bit is set to "1", the HERRn:TOUT bit is also set to "1" at the same time, In the case the read value of this bit is "0", it indicates that no such a error has occurred.. This bit is cleared by writing "0". Writing "1" to this bit is ignored. UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.															
Bit9-8	HS	<p>These bits indicate NULL when handshake operation is not performed due to any error, or when the SOF token triggered by settign the HTOKENn:TKEN bits is completed. These bits are updated when transmission or reception is completed. Handshake values are described below:</p> <table border="1"> <thead> <tr> <th>Bit9</th><th>Bit8</th><th>Handshake</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>ACK</td></tr> <tr> <td>0</td><td>1</td><td>NAK</td></tr> <tr> <td>1</td><td>0</td><td>STALL</td></tr> <tr> <td>1</td><td>1</td><td>NULL</td></tr> </tbody> </table> <p>UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.</p>	Bit9	Bit8	Handshake	0	0	ACK	0	1	NAK	1	0	STALL	1	1	NULL
Bit9	Bit8	Handshake															
0	0	ACK															
0	1	NAK															
1	0	STALL															
1	1	NULL															

62.4.4 Host State Status Register (HSTATEN)

The Host State Status Register (HSTATEN) indicates the status of the USB circuit such as connections to devices and transfer mode.

■ Host State Status Register (HSTATEN)

Figure 62.4-5 Bit Configuration of the Host State Status Register (HSTATEN)

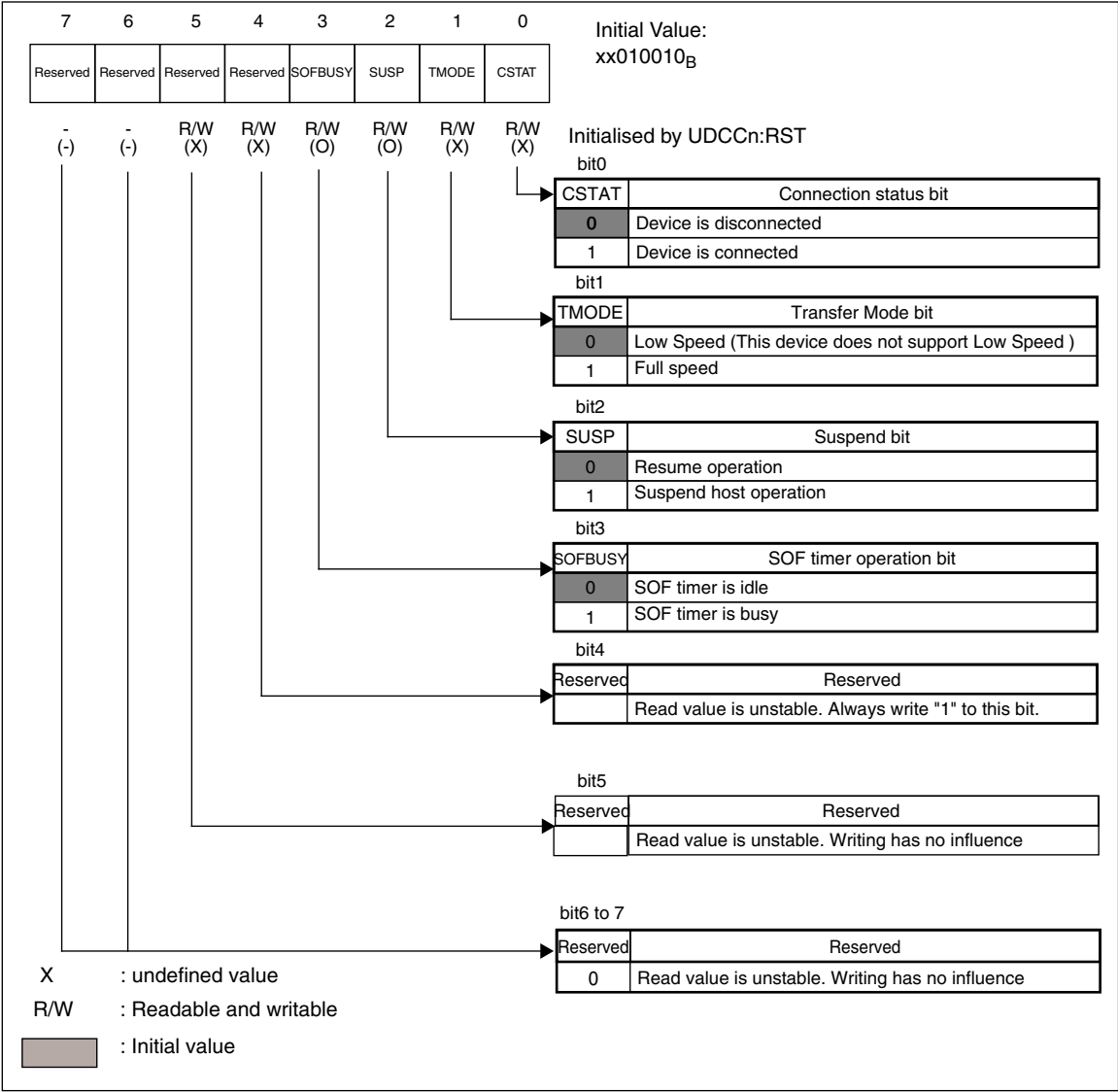


Table 62.4-5 Functional Description of each bit of the HSTATEn registers:

	Bit names	Function								
Bit7-5	Reserved	Read value is unstable. Writing has no influence.								
Bit4	Reserved	Always write "1" to this bit. Read value is unstable.								
Bit3	SOFBUSY	<p>This bit is set to "1" when a SOF token is executed by setting HTOKENn register. When the read value of this bit is "1", it means that the SOF timer is operating. When the read value of this bit is "0" it means that the SOF timer is not operating.</p> <p>Writing "0" to this bit stops SOF timer. Writing "1" to this bit is ignored.</p> <p>It takes time to stop the SOF timer after writing "0" to this bit. To confirm if the SOF timer has been stopped, read this bit.</p> <p>UDCCn:RST bit must be set to "0" to update this bit. When UDCCn:RST bit is "1", this bit is initilaized.</p>								
Bit2	SUSP	<p>This bit sets the suspend status in the host mode.</p> <p>Writing "1" to the bit sets suspend status.</p> <p>Writing "0" to this bit when it is "1" or a change of USB bus to the k-state cancels the suspend status. In this case HIRQn:RWKIRQ is set.</p> <p>It is forbidden to set the bit to "1" while the USB is operating (when resetting the USB bus or sending/receiving data or the SOF timer is operating).</p> <p>In host mode, it is forbidden to stop the USB clock even in suspend status.</p> <p>To update this bit the UDCCn:RST bit must be set to "0".</p> <p>It is prohibited to set this bit to "1" in USB function mode.</p> <p>If it is set to "1" before changing the mode from Host to Function mode, you must get out of suspend status by writing "0" into it before changing the mode. (please refer description below).</p> <table><tr><td>SUSP</td><td>Operation</td></tr><tr><td>Writing "1"</td><td>Suspend</td></tr><tr><td>Writing "0" when it is set to "1"</td><td>Resume</td></tr><tr><td>Other</td><td>State maintenance</td></tr></table>	SUSP	Operation	Writing "1"	Suspend	Writing "0" when it is set to "1"	Resume	Other	State maintenance
SUSP	Operation									
Writing "1"	Suspend									
Writing "0" when it is set to "1"	Resume									
Other	State maintenance									
Bit1	TMODE	<p>This bit indicates the transfer mode, i.e when "1" Full Speed and when "0" Low Speed (This device does not support Low Speed mode).</p> <p>It is not initialized with the UDCCn:RST bit.</p>								
Bit0	CSTAT	<p>This bit indicates whether a device is connected or not, in Host mode. When the read value of this bit is "1", it means that the USB device is connected. When the read value of this bit is "0", it means that the USB device is not connected.</p> <p>This bit is not initialized with the UDCCn:RST bit.</p>								

62.4.5 SOF Interruption FRAME Comparison Register (HFCOMPn)

The SOF interrupt FRAME comparison register (HFCOMPn) is used to set data that is compared with the lower 8 bits of FRAME Number of SOF token.

■ SOF Interruption FRAME Comparison Register (HFCOMPn)

Figure 62.4-6 Bit Configuration of SOF Interruption FRAME Comparison Register (HFCOMPn)

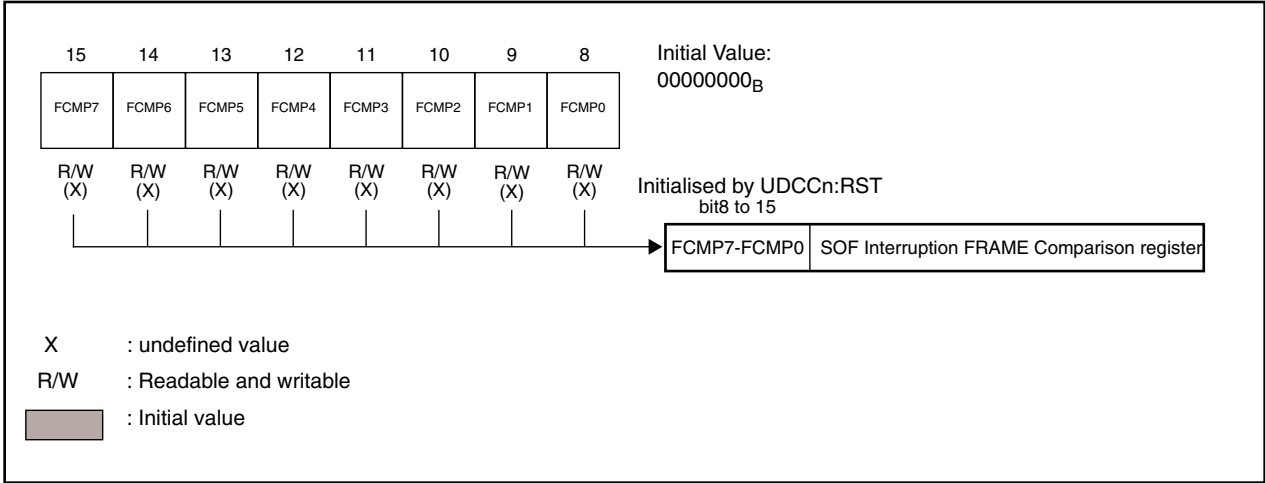
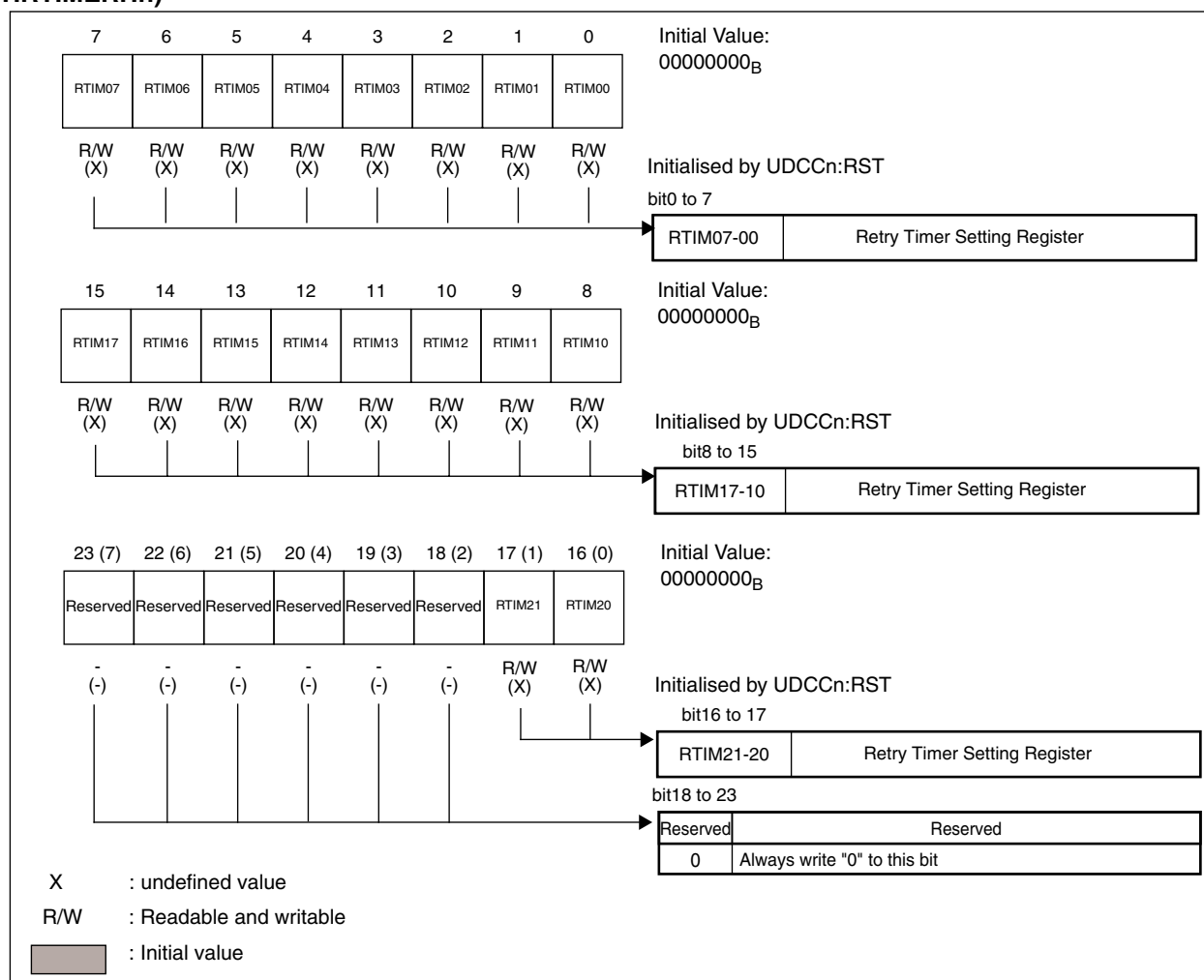


Table 62.4-6 Functional Description of each bit of the HFCOMPn registers:

	Bit names	Function
Bit15-8	FCMP7-FCMP0	<p>This register holds the value to be compared with the lower 8 bits of the Frame Number of SOF token.</p> <p>It is not initialized with the UDCCn:RST bit.</p> <p>When a SOF transmission starts, in the case HCNTNn:SOFSTEP is "0", the FRAME Number in least significant 8 bits of the SOF token and HFCOMP are compared. And if a match is detected, HIRQn:SOFIRQ bit is set. In this case, if HCNTLn:SOFIRE bit is "1", an interrupt is generated.</p> <p>In the case HCNTNn:SOFSTEP is "1", this register is ignored.</p> <p>This register is not initialized with the UDCCn:RST bit.</p>

62.4.6 Retry Timer Setting Registers (HRTIMERLn, HRTIMERMn, HRTIMERHn)

The retry timer setting register (HRTIMERLn, HRTIMERMn, HRTIMERHn) is used to set a retry time period for a token.

■ Retry Timer Setting Register (HRTIMERLn, HRTIMERMn, HRTIMERHn)**Figure 62.4-7 Bit Configuration of Retry Timer Setting Register (HRTIMERLn, HRTIMERMn, HRTIMERHn)****Table 62.4-7 Functional Description of each bit of the HRTIMERLn, HRTIMERMn, HRTIMERHn registers:**

	Bit names	Function
Bit17-0	RTIM21-RTIM20 RTIM17-RTIM10 RTIM07-RTIM00	These bits set the time to retry a token (after receiving a NACK). When the HCNTNn:RETRY bit is "1", a retry timer is activated after the token is started, and the timer is decremented by "1" at the transfer clock rate (12 MHz at Full Speed) until it reaches zero. When the counter reaches zero, the retry operation is finished after the current token is executed. When a token retry occurs in an EOF area, the retry timer stops until the SOF is complete. The down counting is resumed after the SOF is executed This register is not initialized with the UDCCn:RST bit

62.4.7 Host Address Register (HADRn)

The host address register (HADRn) is a register used to set the address field when a token is to be sent.

■ Host Address Register (HADRn)

Figure 62.4-8 Bit Configuration of Host address Register (HADRn)

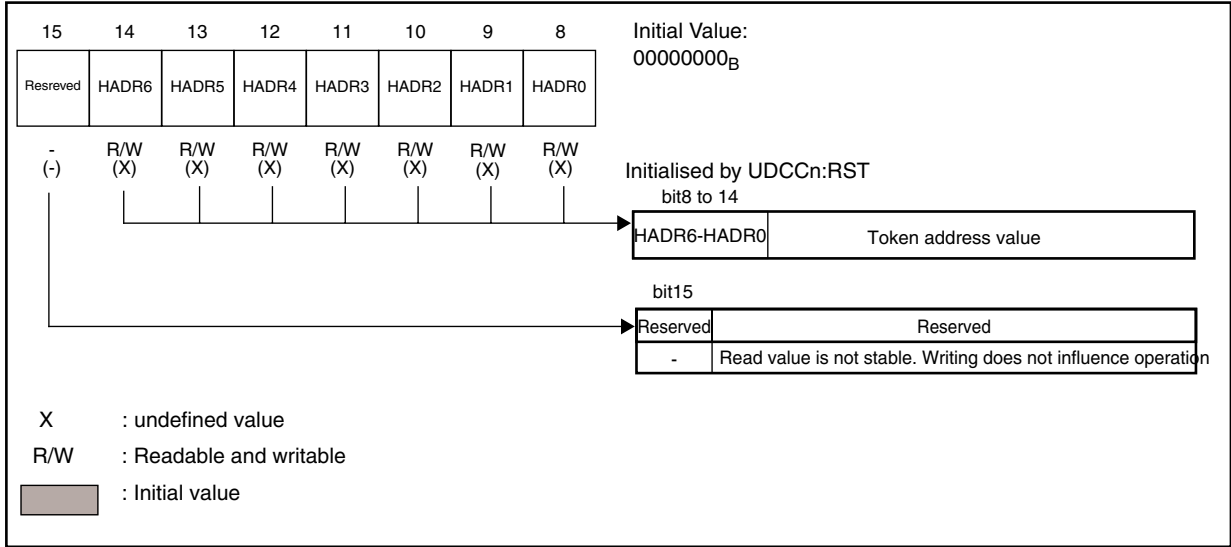


Table 62.4-8 Functional Description of each bit of the HADRn registers:

	Bit names	Function
Bit15	Reserved	Read value is not stable. Writing does not have any influence.
Bit14-8	HADR6-HADR0	Set token address in this register. It is not initialized with the UDCCn:RST bit. This register is not initialized with the UDCCn:RST bit.

62.4.8 EOF Setting Register (HEOFn)

The EOF setting register (HEOFn) defines a time period for which a token is inhibited before the execution of the SOF token.

■ EOF Setting Register (HEOFn)

Figure 62.4-9 Bit Configuration of EOF Setting Register (HEOFn)

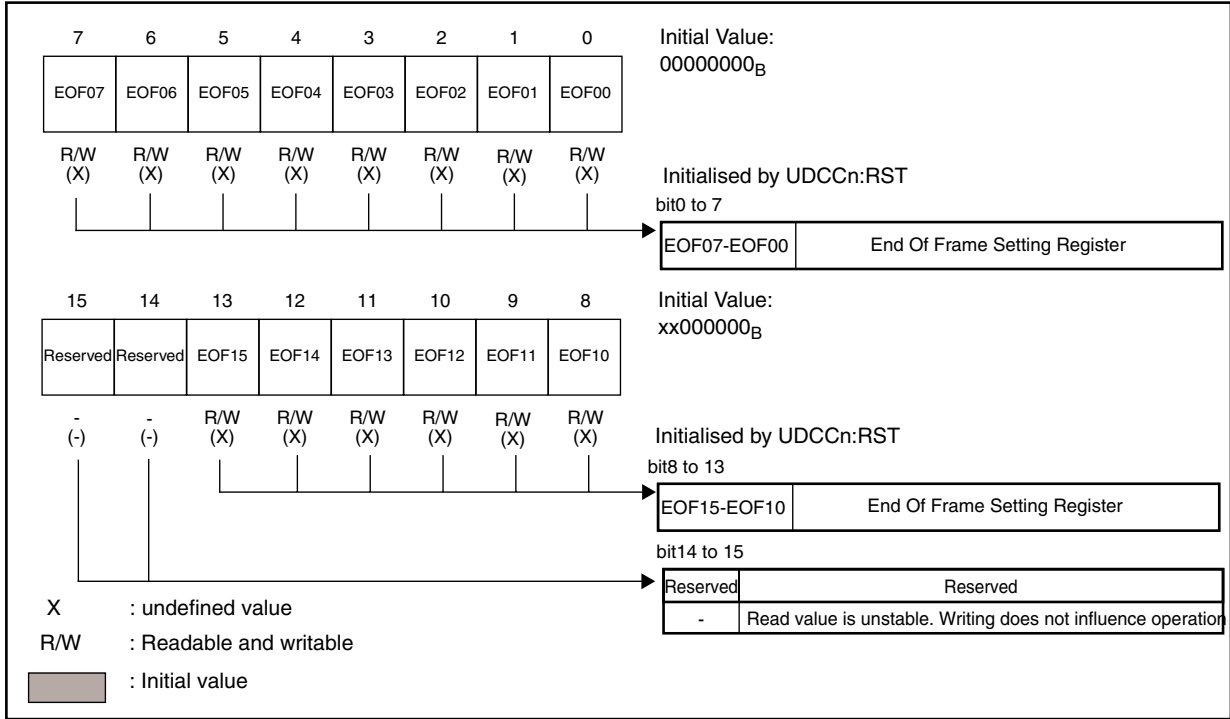


Table 62.4-9 Functional Description of each bit of the HEOFn registers:

	Bit names	Function
Bit15-14	Reserved	Read value is unstable. Writing does not have any influence.
Bit13-0	EOF15-EOF10 EOF07-EOF00	The EOF registers hold a time period during which the execution of a token is inhibited before the execution of SOF. If the value stored in the SOF timer turns out to be lower than value stored in the HEOF register, for any of an IN, OUT or SETUP token execution requests made, they will be run only after the SOF token is executed. This prevents simultaneous execution of any of those tokens with an SOF token. This register is not initialized by setting the UDCCn:RST bit to "1". The time unit of the HEOF register is one bit time. A margin longer than one packet length must be set. e.g. for the MAXPKT=64 byte and Full Speed: (Token_length + packet_length + header + CRC)x7/6 + Turn_around_time = (34bit + 546bit)x7/6 + 36bit = 712.7 bits. As a result, (2C9) _H is to be set in this register.

62.4.9 FRAME Setting Register (HFRAMEn)

The FRAME setting register (HFRAMEn) is used to set the FRAME Number of SOF tokens. SOF is transmitted automatically every 1ms after HTOKENn:TKEN bits are set to SOF transfer (TKEN[2:0] = 100B).

■ FRAME Setting Register (HFRAMEn)

Figure 62.4-10 Bit Configuration of FRAME Setting Register (HFRAMEn)

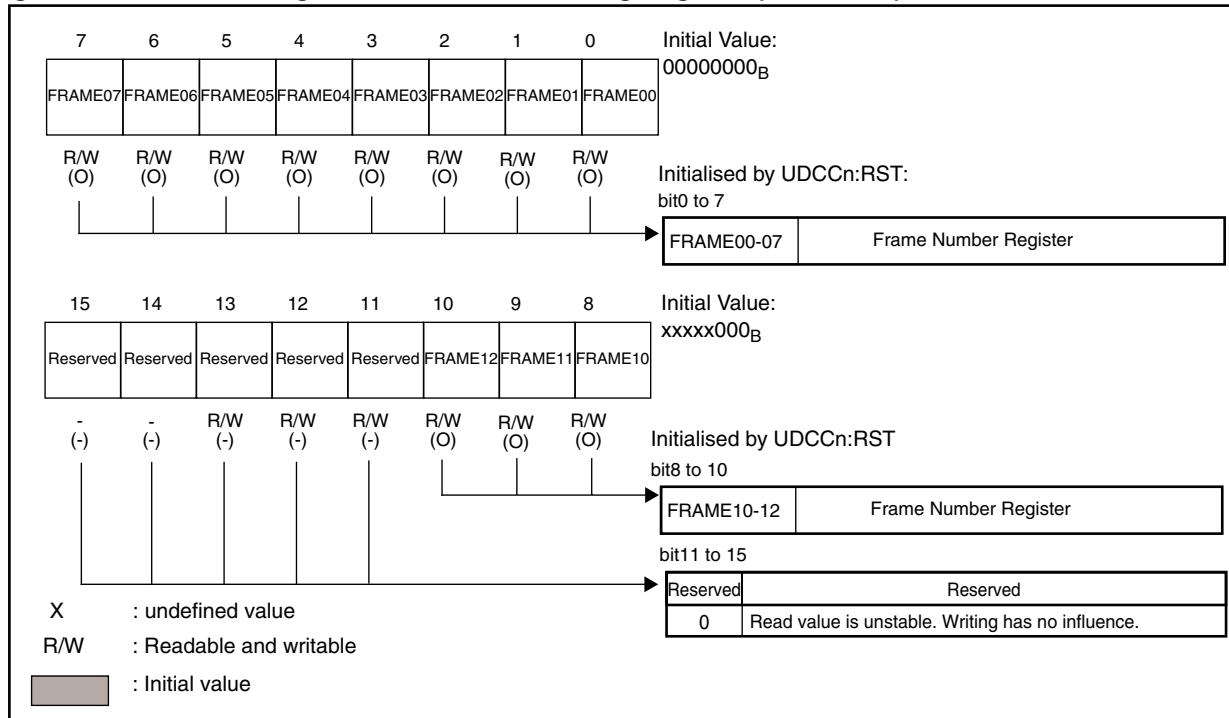


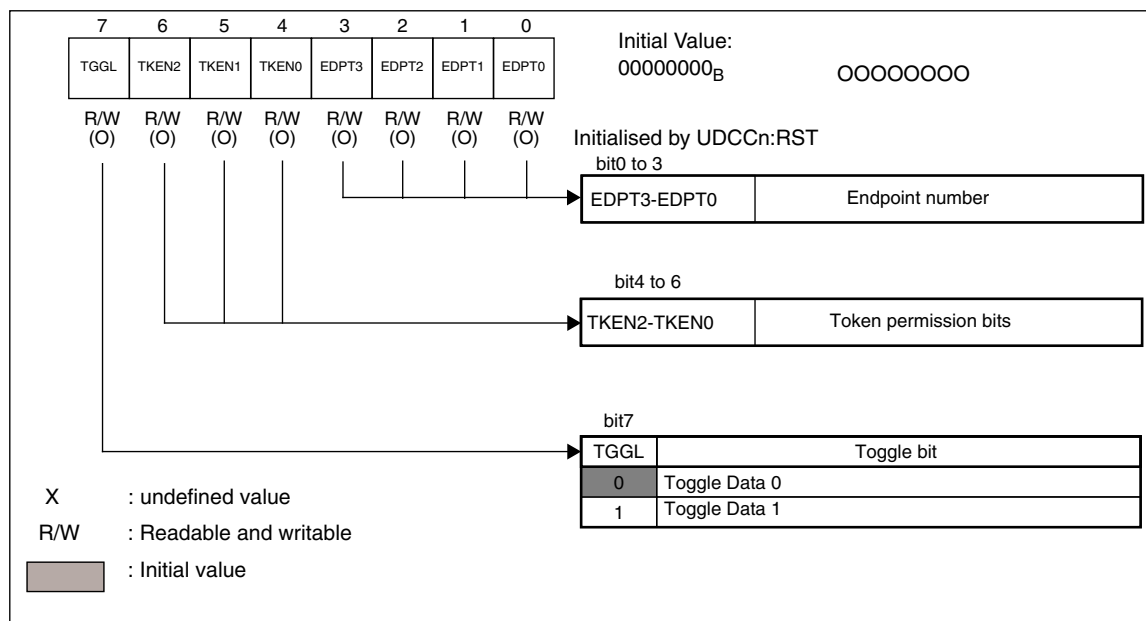
Table 62.4-10 Functional Description of each bit of the HFRAMEn registers:

	Bit names	Function
Bit15-11	Reserved	Read value is unstable. Writing has no influence
Bit10-0	FRAME00 - FRAME07 FRAME10 - FRAME12	<p>These bits are used to set the frame number of SOF token. Before setting the HTOKENn:TKEN bits to SOF (TKEN[2:0] = 100B), the Frame Number must be set to these bits.</p> <p>When the HSTATEn:SOFBUSY bit is "1" or an SOF token is being executed, write operation is inhibited.</p> <p>The UDCCn:RST bit must be set to "0" to update the register. This bit is initialized when UDCCn:RST bit is "1".</p>

When the HTOKENn:TKEN bits to SOF activation are set, the SOF timer starts and, afterwards, an SOF is automatically sent out every 1 ms. The FRAME setting register is automatically incremented by 1 every time a SOF is completed.

62.4.10 Host Token Endpoint Register (HTOKENn)

The host token endpoint register (HTOKENn) is a register that sets a toggle, endpoint, and token.

■ Host Token Endpoint Register (HTOKENn)**Figure 62.4-11 Bit Configuration of Host Token Endpoint Register (HTOKENn)****Table 62.4-11 Functional Description of each bit of the HTOKENn registers:**

	Bit names	Function
Bit7	TGGL	The bit setting defines, at transmission, the toggle value to be sent. At reception, the received toggle data is compared to the toggle data which this bit shows and used for error detection. When updating this bit confirm that UDCCn:RST bit is set to "0" and the TKEN bits are set to "000".
Bit6-4	TKEN2-TKEN0	Setting these bits sends a token corresponding to its value. Once the operation is completed, these bits are changed to "000B", and the HIRQn:CMPIRQ bit is set to "1". If HCNTLn:CMPIRE bit is set an interrupt is generated. When update TKNEN bits, the UDCCn:RST bit must be "0". These bits are initialized when UDCCn:RST bit is "1". and operation mode is to be Mini-Host mode. If a token is issued again after a interrupt due to token is generated, you should wait for three cycles or longer in terms of USB transfer clock (12 MHz) before writing to the TKNEN bits. Writing to the TKNEN bits will not transmit any token in disconnection status (HSTATEn:CSTAT = "0"). In the case that the SOF token is set to these bits (100B), setting of TGGL bit and ENDPT bit are ignored. If the HSTATEn:SOFBUSY bit is set, writing "100" to these bits is forbidden. (Please refer to "Table 62.4-12 Token Setting")
Bit3-0	EDPT3-EDPT0	Those bits define the transmit and receive endpoint of the device. The UDCCn:RST bit must be set to "0" to update the register.

Note: The TGGL and EDPT bits are ignored when a SOF token is executed.

Table 62.4-12 Token Setting

Bit6	Bit5	Bit4	Operation
0	0	0	No token sent out
0	0	1	SETUP token sent out
0	1	0	IN token sent out
0	1	1	OUT token sent out
1	0	0	SOF token sent out

62.5. USB Mini-host operation

This section describes the operation of USB Mini-Host.

62.5.1 Device Connection

The method of detecting the connection of an external USB device by software is described.

■ Setting of Mini-host Function

The HCNTLn:HOST bit must be set to "1" to turn the USB function as a host.

■ Disconnection Status, Connection Status of an External USB Device

When the external USB device is disconnected, both pins UDP and UDM, are "L" because of the pull-down resistors. Then the HSTATEn:CSTAT bit is "0", and the TMODE bit is undefined. The HSTATEn:CSTAT bit becomes "1" when the external USB device is connected.

■ Detection the connection of an external USB device

When an external USB device is connected, the HIRQn:CNNIRQ bit becomes "1". In the case that the HCNTLn:CNNIRE bit is "1", a device connection interrupt is generated. Clearing this bit to clear the interrupt. To detect the connection of the external USB device not through interrupt but through polling, a program so that it ensures that the HCNTLn:CNNIRE bit is set to "0" and then HIRQn:CNNIRQ bit is set to "1".

■ Acquiring transfer speed of destination USB device and selecting the clock

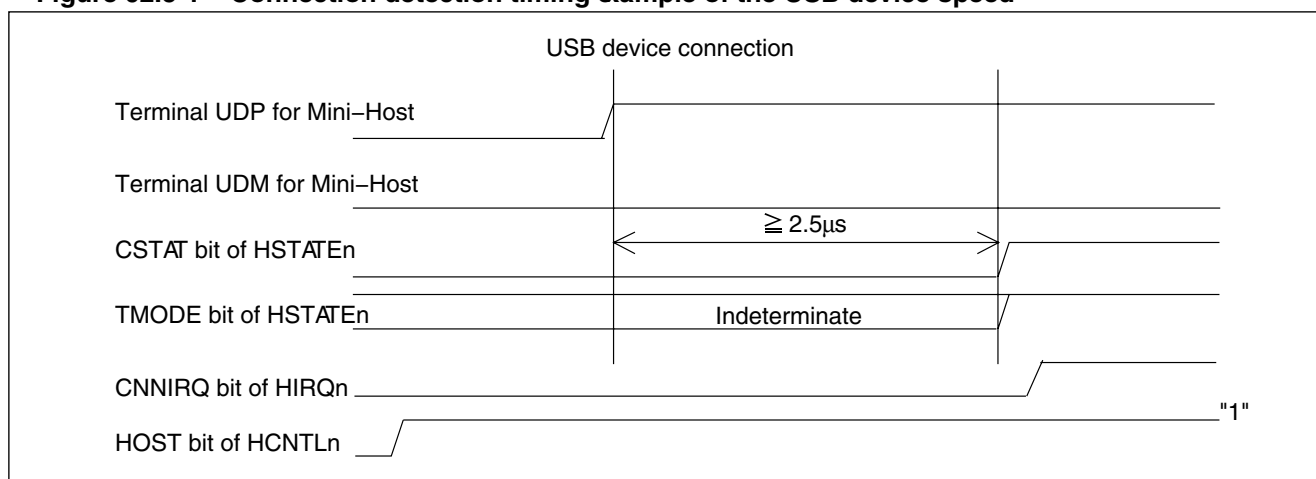
HSTATEn:TMODE bit holds the information regarding the transfer speed of the connected USB device.

If the connected device is a Full Speed device then TMODE = "1" .

If the connected device is a Low Speed device then TMODE = "0" .

Note: Low Speed mode is not supported by this device.

Figure 62.5-1 Connection detection timing example of the USB device speed



Note: The HSTATEn:CSTAT bit is set to "1" in 2.5 μs after the external USB device is connected. The HSTATEn:TMODE and HSTATEn:CSTAT bits are updated regardless of the setting of the HCNTLn:HOST bit.

62.5.2 USB Bus Reset

When the HCNTLn:URST bit is set to "1" in the host mode, USB Mini-Host drives SE0 for not less than 10 ms to indicate a reset condition to the device. When the USB bus reset is completed, the URST bit is cleared to "0" and HIRQn:URIRQ bit is set to "1". In this case, if the HCNTL: URIRE bit is enabled, an interrupt is generated. Clearing HIRQn:URIRQ bit clears the interrupt.

● Precautions to be taken when resetting the USB bus

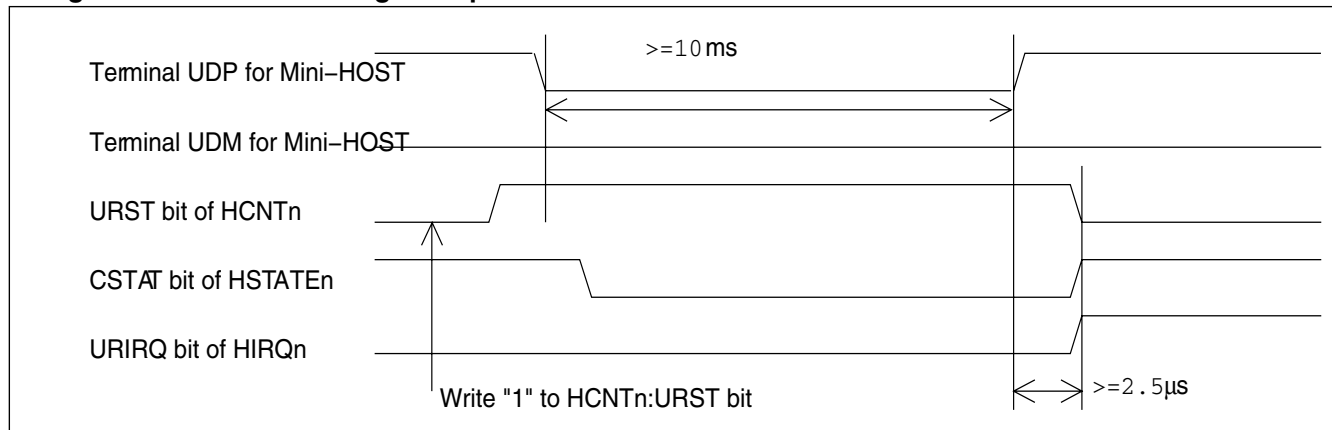
Please note the following when resetting the USB bus:

Before the USB bus is reset, HSTATEn:CSTAT bit must be checked to confirm if the device is connected.

When the USB bus is being reset, the HSTATEn:CSTAT bit is automatically cleared to "0" and the USB device status turns "disconnected". In this case the HIRQn:DIRQ bit is not set to "1".

After the USB bus reset is completed, the HSTATEn:TMODE bit should be checked to confirm that the connected device is "Full Speed" device.

Figure 62.5-2 Reset Timing Example to Device



62.5.3 Token Packet

To execute any of IN, OUT or SETUP token, the token packet is started when the necessary data are set in the Host Token Register (HTOKENn) after setting the HADRn register, DIR bit of EP1Cn register (or EP2Cn register) and PKS bit of EP1Cn register (or EP2Cn register). In case you want to send a SOF token, you must set necessary data in the Host Token Endpoint Register (HTOKEN) after configuring the FRAME Setting Register (HFRAME) and EOF Setting Register (HEOF). If registers (HADRn, EP1Cn, EP2Cn, HFRAMEn and HEOFn) have not been changed, it is not required to set them.

■ Setting of Token Packet

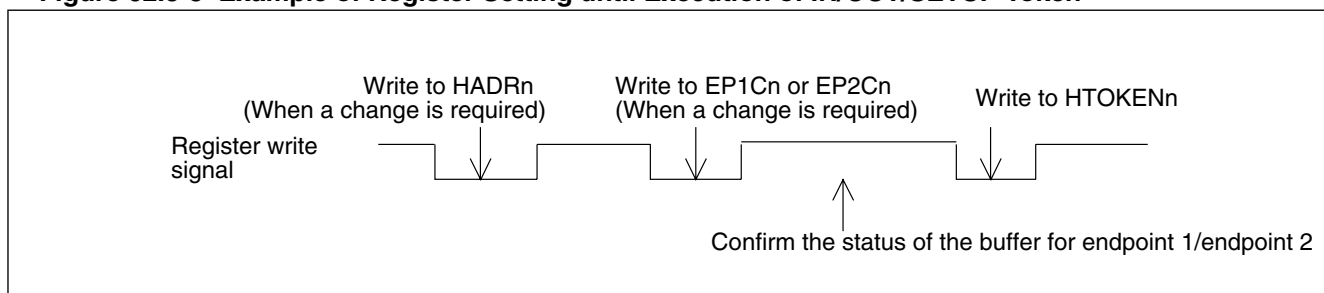
In the host mode endpoint 1 and endpoint 2 are used as transmission buffer and reception buffer.

For the IN, OUT or SETUP token, set the destination address in the host address register (HADRn), and set the maximum transferable number of bytes in one packet is set in the EP1Cn:PKS bits (or the EP2Cn:PKS bits), and the the direction of the token transfer is to be set in the EP1Cn:DIR bit (or the EP2Cn:DIR bit). If the EP1Cn:DIR is set to "1", the buffer for endpoint 1 is used for OUT-direction transfer (transmit buffer) and the buffer for endpoint 2 is used for IN-direction (receive buffer). In this condition, the EP2Cn:DIR bit must be set to "0" (The EP1Cn:DIR and EP2Cn:DIR bits should be complement of each other i.e, if one is IN buffer the other should be OUT buffer). On similar lines, when EP1Cn:DIR bit is "0", the buffer for endpoint 1 is used for IN-direction transfers and the buffer for endpoint 2 is used for OUT direction transfers. In this condition, the EP2Cn:DIR bit must be set to "1".

To excute a TOKEN, the setting is to be done in following sequence.

1. The direction is to be specified in to the DIR bits of EP1Cn and EP2Cn.
2. In the case that the target endpoint m (m=1 or 2) is for OUT transfer, write the transmission data in to the buffer of endpoint m, and clear the EPmSn:DRQ bit to "0". In the case the direction is IN transfer, readn the EPmSn:DRQ bit to confirm that it is "0".
3. Set the target endpoint, token and toggle data in to the host token endpoint register (HTOKENn).rgct endpoint token and toggle data in the Host Token endpoint register (HTOKENn).

The USB circuit sends out a token packet in the following order: Sync, token, address, endpoint, CRC5, and EOP based on the specified token (a Sync, CRC5, and EOP are automatically sent). Once one packet is complete, the HIRQn:CMPIRQ bit is set to "1", and the HTOKENn:TKEN bit is set to 000_B (See "62.5.7 SOF Interrupt"). If the HCNTLn:CMPIRE bit is "1" at this time, an interrupt occurs. Clearing HIRQn:CMPIRQ clears the interrupt.

Figure 62.5-3 Example of Register Setting until Execution of IN/OUT/SETUP Token

In the case of an SOF token, when an EOF time and FRAME number are set in the EOF setting register (HEOFn) and FRAME setting register (HFRAMEn) respectively, and the SOF token code to the HTOKENn:TKEN bits is written, a Sync, SOF token, FRAME number, CRC5, and EOP are sent out and the HSTATEn:SOFBUSY bit is set and the HFRAMEn is incremented. The HIRQn:CMPIRQ is also set (triggering an interrupt if enabled in the HCNTn:CMPIRE bit) and the HTOKENn:TKEN bits are cleared. Next or later SOF executed automatically does not cause an interrupt by the HIRQn:CMPIRQ.

SOF is automatically sent out every 1 ms as long as the HSTATEn:SOFBUSY bit is set to "1".

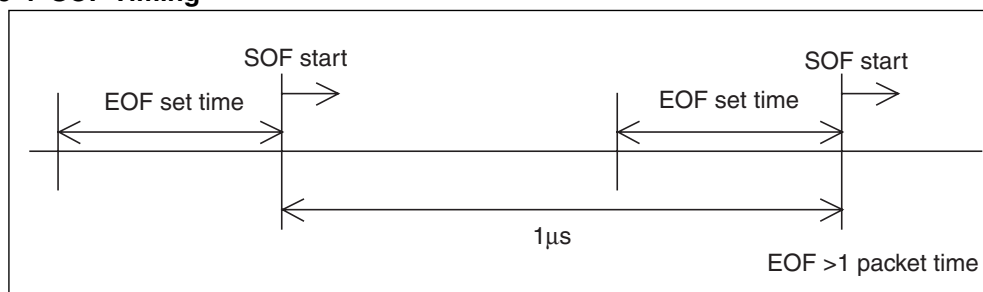
The conditions (SOF stop conditions) that set the HSTATEn:SOFBUSY bit to "0" are described below:

- Writing "0" to HSTATEn:SOFBUSY bit directly.
- A USB bus Reset (Writing "1" to HCNTn:URST).
- A HSTATEn:SUSP bit is set (by a direct write access).
- Disconnection of the device (HSTATEn:CSTAT bit gets cleared to "0").

To switch from host mode to device mode, at first clear the HSTATEn:SOFBUSY bit to "0", confirm that the read value of HSTATEn:SOFBUSY is "0" and read value of the HTOKENn:TKEN bits are "000B" and read value of the HSTATEn:SUSP is "0", and then clear HCNTLn:HOST bit to "0" to change the mode to device mode.

If you want to set back the HSTATEn:SOFBUSY bit to "1" again, you need to run an SOF token once again.

To prevent the simultaneous executions of a SOF token with other tokens, the EOF setting register is used to send a token (which is already set) after waiting for SOF completion, if the HTOKENn:TKEN bits are written between the EOF setting time and the SOF start time. The unit of time for the EOF setting register is one bit time. For example, when setting 10_H to the EOF setting register, the following time is required: $16 \times 1/12 \text{ MHz} = 1333.3 \text{ ns}$ in Full speed mode. If the EOF set time is shorter than one packet time, the SOF execution may overlap other token execution. In this case, the HERRn:LSTOF bit is set to "1" and the SOF is not executed. When the HERRn:LSTOF bit is set to "1", the data of the EOF setting register must be increased (See "62.4.8 EOF Setting Register (HEOFn)").

Figure 62.5-4 SOF Timing

62.5.4 Data Packet

In case a data packet is transmitted after a token packet has been sent, a toggle data is transmitted based on the HTOKENn:TGGL bit, and then the buffer data (from the transmit buffer of EndPoint1 or EndPoint2 that is selected according to the EP1Cn:DIR bit setting), the CRC16 data, and EOP is sent. In case of receiving the data packet, the HTOKENn:TGGL bit and the received toggle data are compared and in case of match, the received data is stored in the buffer of endpoint 1 or endpoint 2 depending on the EP1Cn:DIR bit setting and the CRC16 error is checked.

■ Data Packet

After sending a token packet, the data packet is executed in the following procedure:

● At Transmission

- Sync is sent automatically.
- DATA0 is transmitted when the HTOKENn:TGGL bit is "0" and DATA1 is transmitted when the HTOKENn:TGGL bit is "1".
- When the EP1Cn:DIR is set to "1", the buffer for endpoint 1 is selected, otherwise endpoint 2 buffer is selected and all data is transmitted.
- EP1Cn:DIR bit is to be set to inverted value of EP2Cn:DIR bit, when the HCNTLn:HOST bit is "1". For example, when the EP1Cn:DIR bit is "0", the EP2Cn:DIR bit is set to "1".
- The CRC16-bits are sent.
- The EOP 2-bits are sent.
- The J State 1-bit is sent.

● At Reception

- Sync is received
- The toggle data is received and is compared with the HTOKENn:TGGL bit.
- In the case they match, if EP1C:DIR bit ="0" the received data is stored in the buffer for endpoint 1 otherwise it is stored in the buffer of endpoint2.
- When the EOF is received, the CRC16 bit is checked.
- EP1Cn:DIR bit is to be set to inverted value of EP2Cn:DIR bit, when the HCNTLn:HOST bit is "1". For example, when the EP1Cn:DIR bit is "0", the EP2Cn:DIR bit is set to "1".

62.5.5 Handshake Packet

Handshake packet is used to inform the status of Mini-Host to the USB device.

■ Handshake Packet

The handshake packet is used by the receiver to inform from the receiver to the transmitter about the status of the transmission. The receiver transmits one of ACK, NAK or STALL status in a handshake packet to inform the transmitter whether it is in proper condition and can receive data properly. When the USB circuit receives a handshake packet, the type of received handshake packet is set in the HERRn:HS bits. When the handshake packet is transmitted, the type of transmitted handshake packet is set in the HERRn:HS bits.

62.5.6 Retry Function

This function enables the transfer error handling by retransmission.

■ Retry Function

If NAK or an error such as CRC error occurs when the packet transfer is completed, USB Mini-Host continues to retry the transfer during a time period set in the retry timer register (HRTIMERLn, HRTIMERMn, HRTIMERHn) if the retry function is enabled (i.e. HCNTHn:RETRY is set to "1").

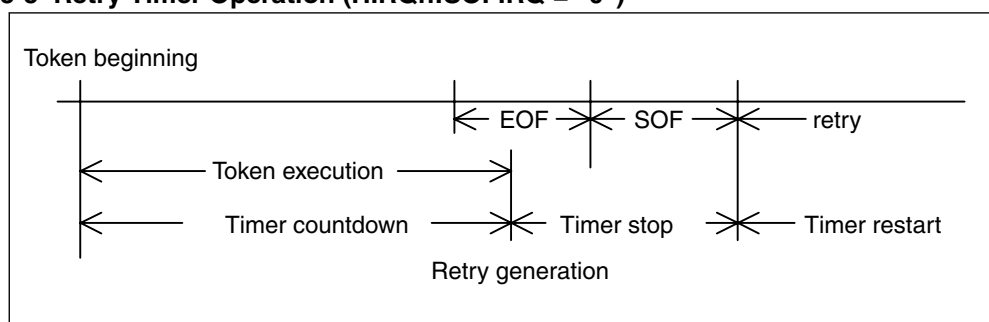
If an error except STALL and disconnected device happens (HERR:HS="01", HERR:RERR="1", HERR:TOUT="1", HERR:CRC="1", HERR:STUFF="1"), USB Mini-Host retries to process the token as long as the HCNTHn:RETRY bit is set to "1".

The retry is stopped by the following conditions:

- HCNTHn:RETRY bit is cleared to "0".
- Detecting "0" in the retry timer registers .
- The occurrence of an interrupt due to SOF (HIRQn:SOFIRQ = "1")
- ACK Detection
- Detection of device disconnection

The retry timer is activated when the token processing starts, counts down with one-bit transfer clock and stops counting when a retry happens in an EOF area. If HIRQn:SOFIRQ bit is "0" when a SOF token is completed, the retry timer restarts from the last value hold in the counter when it was stopped. When the SOF token is complete and the retry timer is "0", the packet is terminated and then HIRQn:CMPIRQ bit is set to "1".

Figure 62.5-5 Retry Timer Operation (HIRQn:SOFIRQ = "0")



When the retry operation is completed, end information on the complete packet is set in related registers.

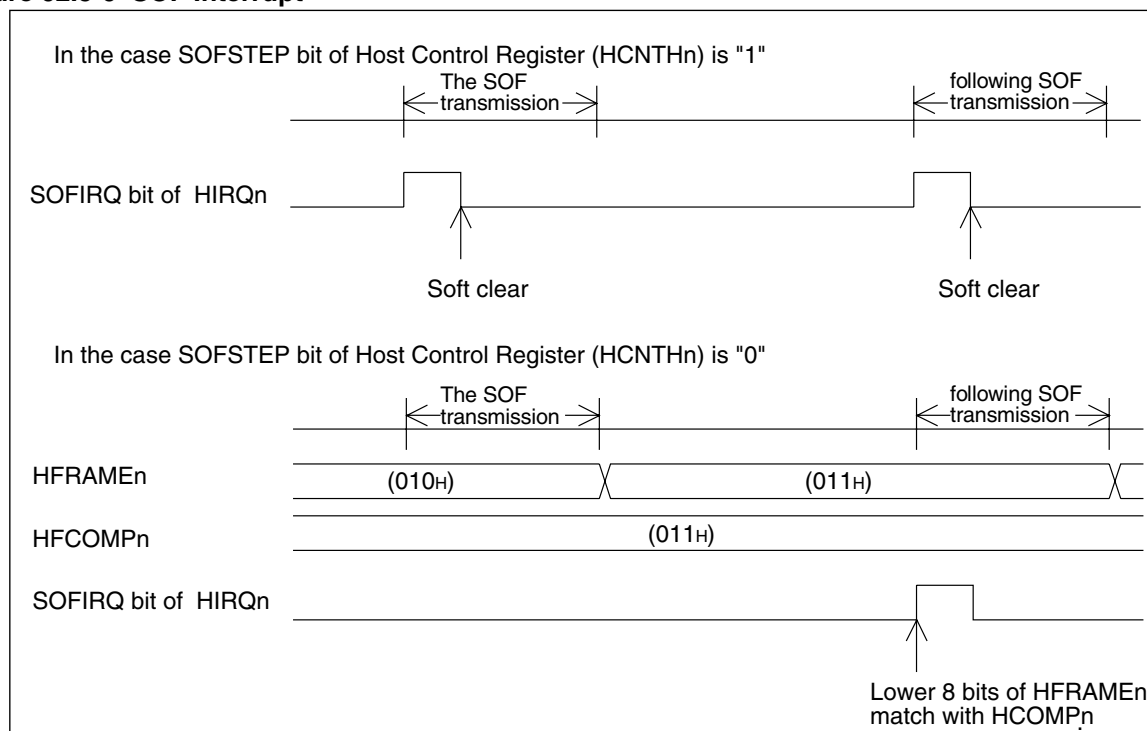
62.5.7 SOF Interrupt

This chapter describes the SOF interrupt processing.

■ SOF Interrupt

When a SOF starts, HCNTLn:SOFIRQ bit is set to "1" according to HCNTLn:SOFSTEP bit and the SOF interruption FRAME comparison register (HFCOMPn) settings. If the HCNTLn:SOFIRE bit is set to "1", an interrupt is triggered when HIRQn:SOFIRQ bit is set. If SOFSTEP is "0", SOF Interrupt Request Flag HIRQn:SOFIRQ bit is set when the lower 8 bits of SOF Frame number are equal to the value set in the SOF Interruption Frame Comparison Register(HFCOMPn). If SOFSTEP is "1", SOF Interrupt Request Flag HIRQn:SOFIRQ bit is set whenever SOF is executed. But HIRQn:SOFIRQ bit is not set by the first SOF token (the SOF token is set by setting the HTOKEN:TKEN bits). The interrupt is enabled if HCNTLn:SOFIRE bit is set. The first SOF execution with the host token endpoint register (HTOKENn) setting does not set the HIRQn:SOFIRQ bit to "1".

Figure 62.5-6 SOF Interrupt



If you set the HCNTLn:CANCEL bit and a token other than SOF (in the HTOKENn) is set in the EOF area, and the HIRQn:SOFIRQ bit is set to "1" in the next SOF, the token required by HTOKENn is not executed and the HTOKENn:TKEN bits are set to 000_B. In this case, the HIRQn:CMPIRQ bit is not raised.

Cancellation of a token can be known by checking the HIRQn:TCAN (Token Cancellation) flag after the HIRQn:SOFIRQ flag is raised. If the token is to be executed again, HIRQn:TCAN flag must be cleared, and the token settings must be specified again in the HTOKENn:TKEN field.

If the HCNTLn:CANCEL bit is set to "0", the token set in the HTOKENn register is executed after the SOF is sent.

Figure 62.5-7 Example of Token Cancel Operation when HCNTn:CANCEL bit is "1".

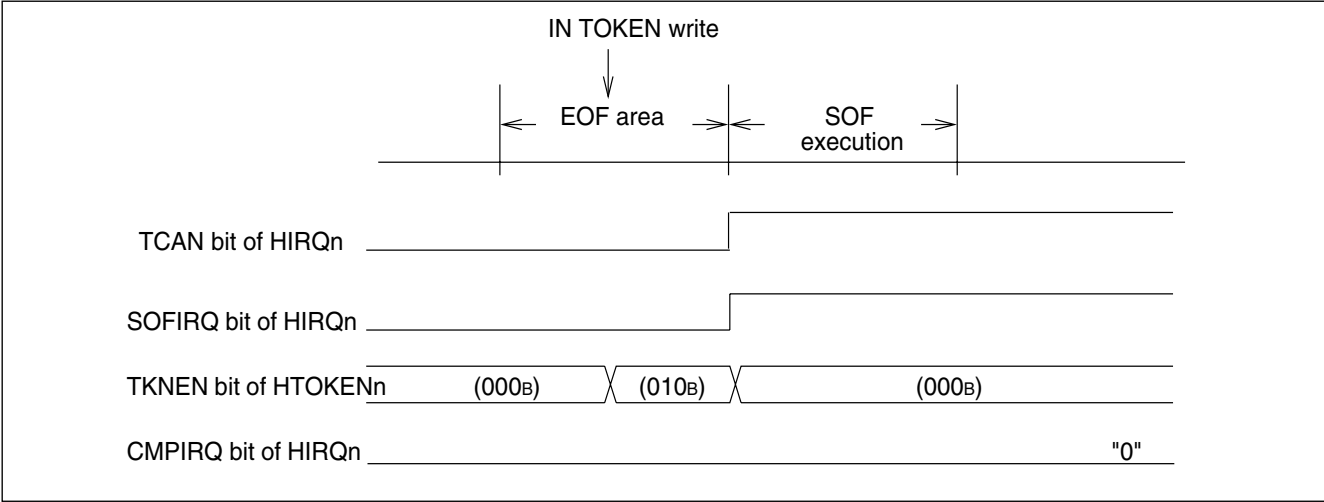
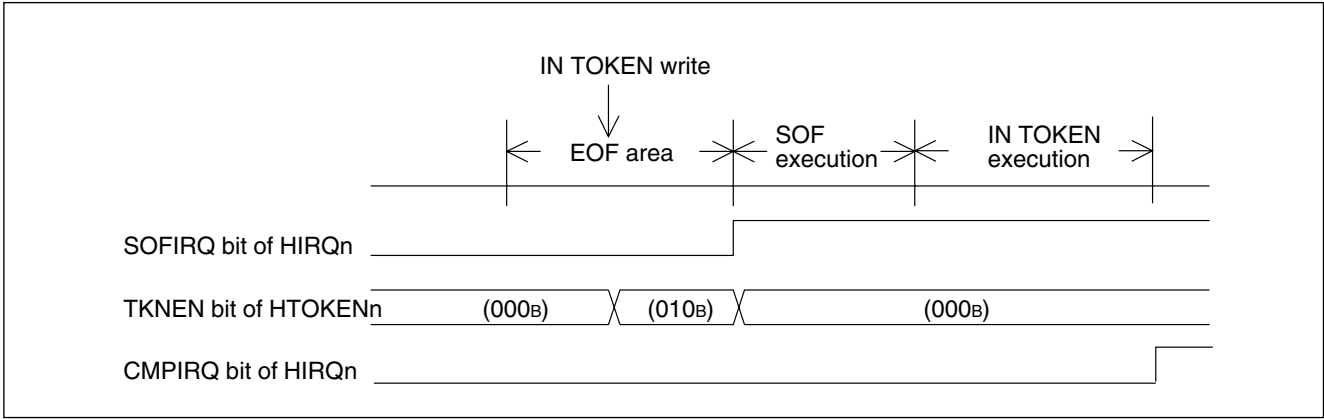


Figure 62.5-8 Example of Token Operation when HCNTn:CANCEL bit is "0".



62.5.8 Error Status

This chapter describes the various error types that the USB Mini-host can detect.

■ Error Status

● Stuffing error

If 6 bits are continuously equal to "1", one "0" bit must be stuffed in the sequence to keep the receiver synchronized. If 7 bits in a row are equal to "1" are encountered, the HERRn:STUFF bit is set signaling a stuffing error. Write "0" to clear HERRn:STUFF bit. If the next token is executed without clearing the STUFF bit, it is updated when the next token is completed.

● Toggle error

When an IN token is received, the HERRn:TGERR bit is set if the toggle data for the data packet and the HTOKENn:TGGL bit do not match. Write "0" to clear HERRn:TGERR bit. If the next token is executed without clearing the TGERR bit, it is updated when the next token is completed.

● CRC error

When an IN token is received, a CRC calculation is performed on the received data with the CRC polynomial $G(X) = X^{16} + X^{15} + X^2 + 1$. If the remainder is not 800D_H, then a CRC error is detected and the HERRn:CRC bit is set. Write "0" to clear HERRn:CRC bit. If the next token is executed without clearing the CRC bit, it is updated when the next token is completed.

● Time-out error

The HERRn:TOUT bit is set if a data packet or handshake is not received within a the specified interval or if SE0 is detected in the received data, or a stuffing error is detected. Write "0" to clear HERRn:TOUT bit. If the next token is executed without clearing the TOUT bit, it is updated when the next token is completed.

● Receive error

The HERRn:RERR bit is set if the received data size is greater than the packet size limit. The packet size limit specified in the EP1Cn:PKS field is referred when endpoint 1 is used and EP2Cn:PKS is referred when endpoint 2 is used. Write "0" to clear HERRn:RERR bit. If the next token is executed without clearing the RERR bit, it is updated when the next token is completed.

62.5.9 Packet End

This chapter describes the Packet End process.

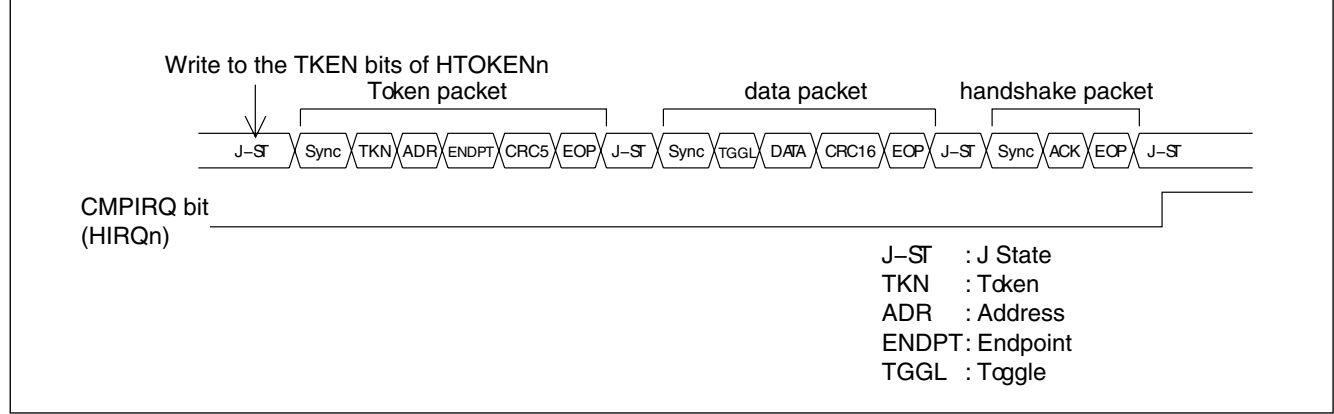
■ Packet End Timing

When one packet transfer is completed in USB Mini-host, an interrupt is triggered by the HIRQn:CMPIRQ bit (if the interrupt is enabled by setting HCNTLn:CMPIRE).

The interrupt is generated in the following timing:

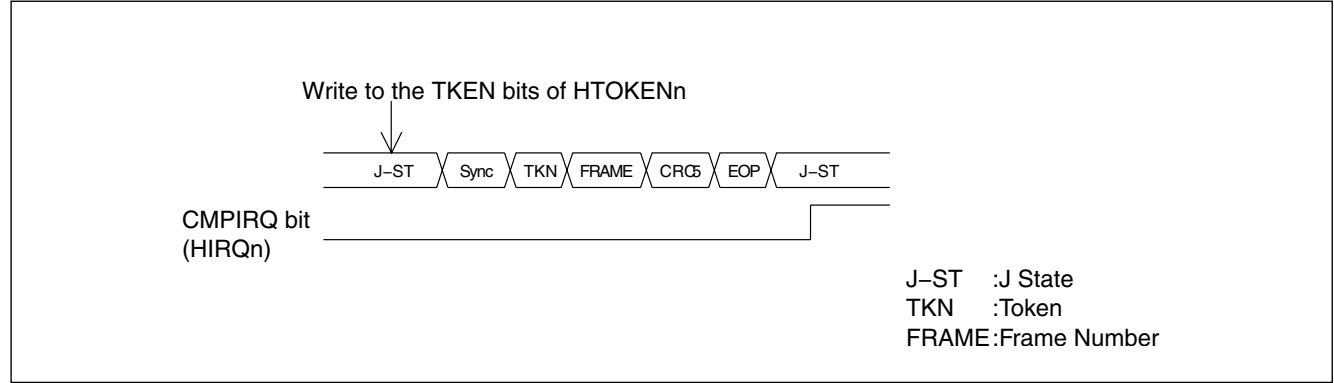
- When the HTOKENn:TKEN bits are 001_B, 010_B or 011_B (SETUP token, IN token, and OUT token)

Figure 62.5-9 First example of HIRQn:CMPIRQ Bit Set Timing



- When the HTOKENn:TKEN is 100_B (SOF token)

Figure 62.5-10 Second example of HIRQn:CMPIRQ Bit Set Timing (SOF TOKEN)



62.5.10 Suspend Resume

USB Mini-host supports suspend and resume operations.

■ Suspend Operation

When "1" is written to the HSTATEn:SUSP bit, USB Mini-Host follows steps below and transits to suspend status.

- USB bus is set in a high impedance state.
- Circuit blocks where clock is not necessary is stopped.

When the USB circuit is put in suspend status, it sets the SUSP bit of the host status register (HSTATEn:SUSP) to "1". It is prohibited to set HSTATEn:SUSP to "1" when USB bus is being reset or the HSTATEn:SOFBUSY bit is "1" or data is being sent or received. Stopping clock supplied to the USB circuit is also forbidden in such case.

■ Resume Operation

Resume operation is started when one of the following conditions is fulfilled:

Writing "0" to HSTATEn:SUSP bit.

The pins UDP and UDM for Mini-host are detected to be in k-state.

Detecting device disconnection.

Detecting device connection.

After the HIRQn:RWKIRQ bit is set, token is allowed again. The followings diagram s show the operation timing for each condition:

Figure 62.5-11 Resume Operation by Register (Full Speed Mode)

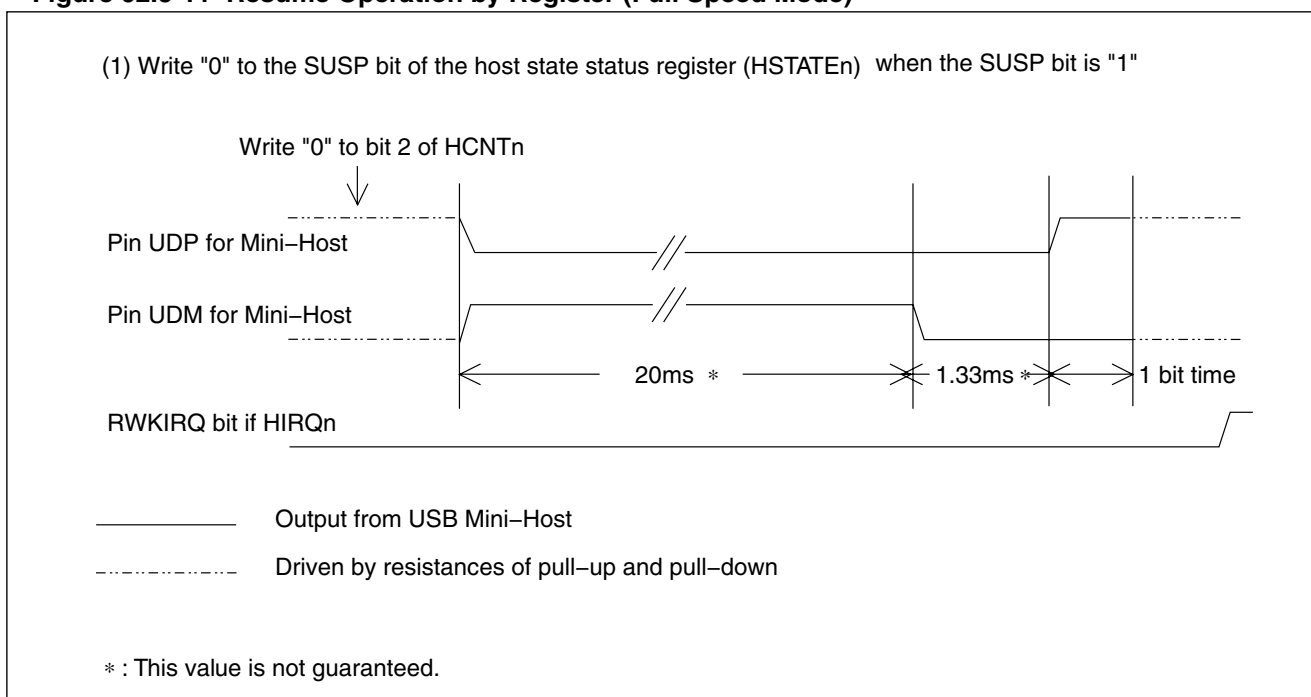


Figure 62.5-12 Resume Operation by detecting K-state on UDP and UDM (Full Speed Mode)

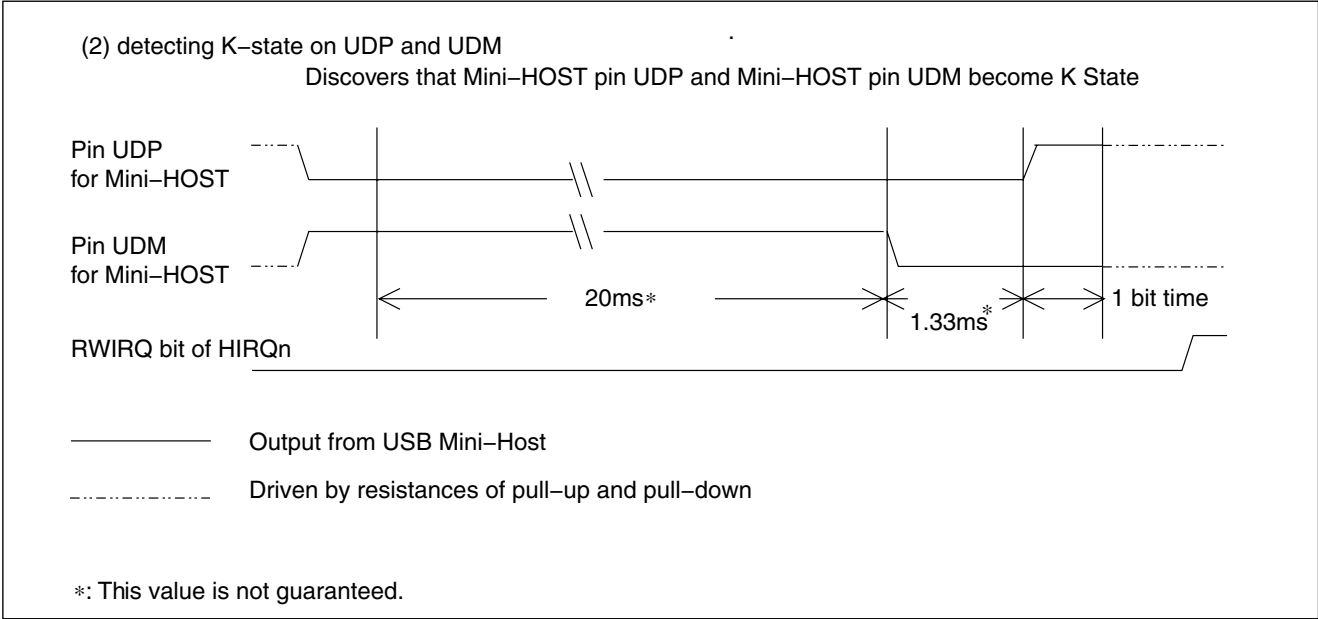


Figure 62.5-13 Resume Operation by Device disconnection

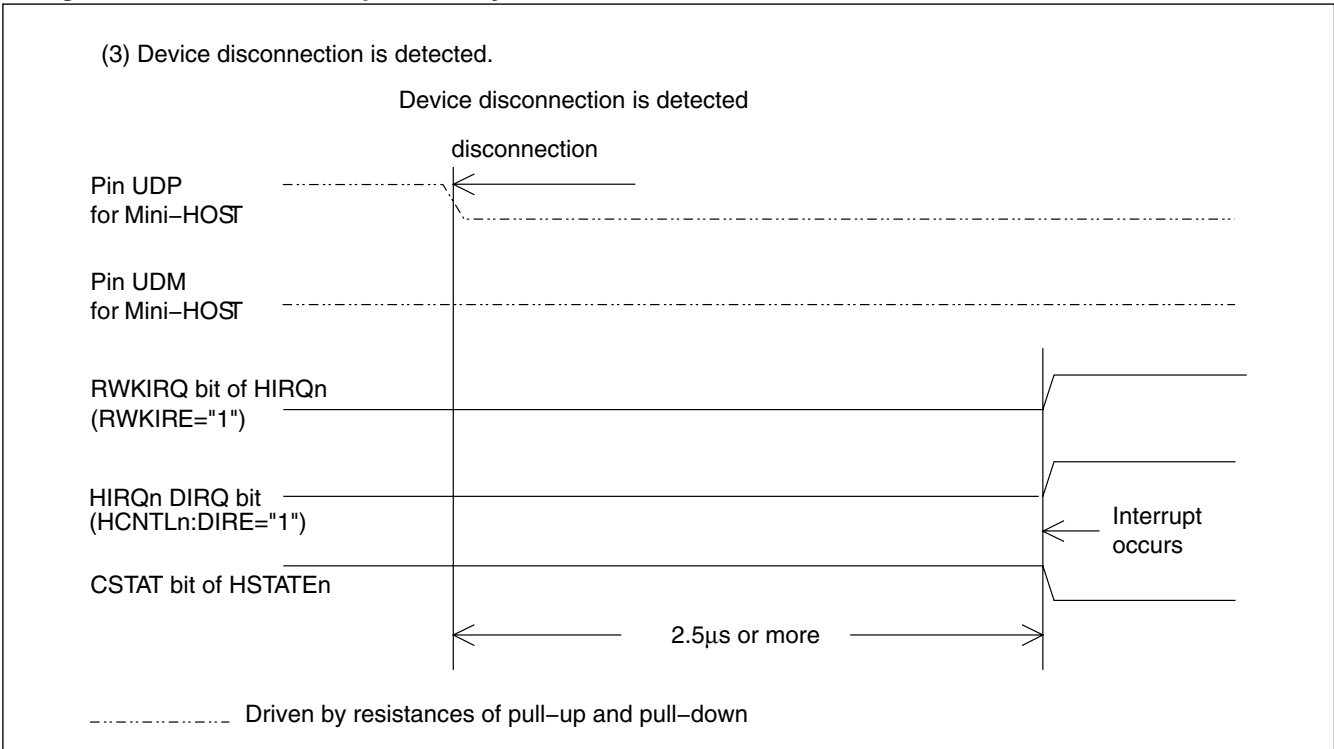
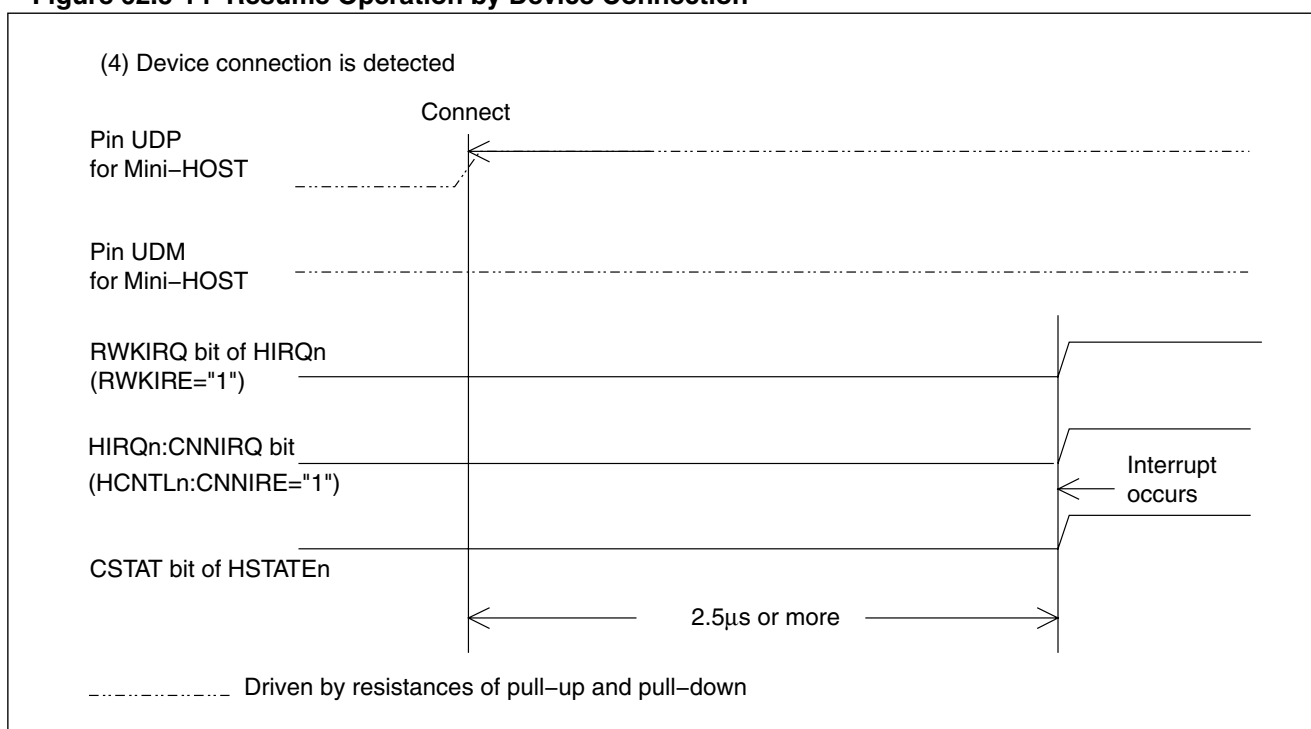


Figure 62.5-14 Resume Operation by Device Connection

62.5.11 Device disconnection

Once both Mini-host pins UDP and UDM become "L", the disconnection timer starts, and sets the HSTATEn:CSTAT bit to "0" when both pins detect "L" for 2.5 µs or longer.

■ Device disconnection

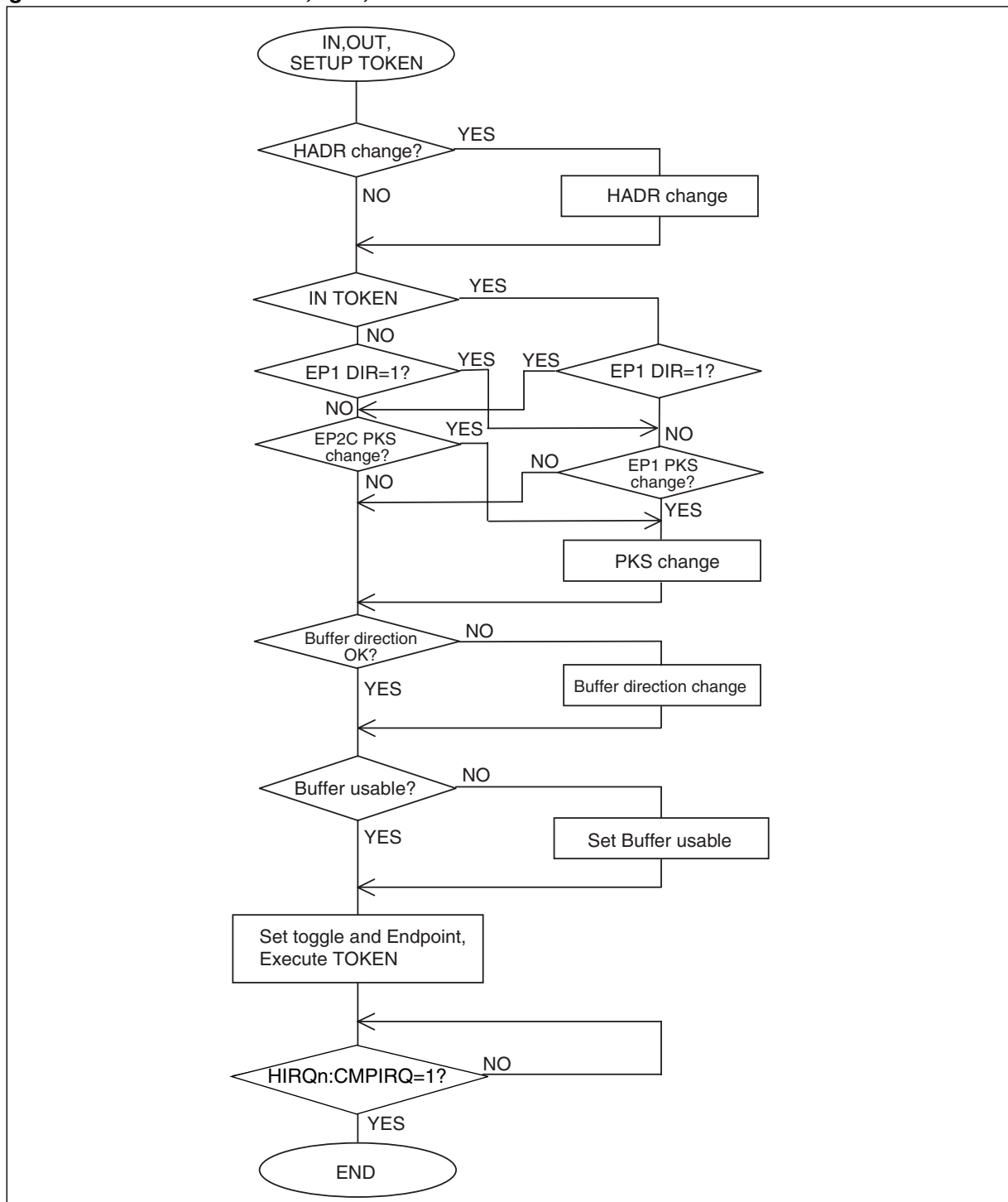
When both mini-host pins UDP and UDM detect "L" for 2.5 µs or longer, it determines that the device is disconnected. Therefore, the HSTATEn:CSTAT bit becomes "0" and the HIRQn:DIRQ flag is also set to "1". If the HCNTLn:DIRE bit is "1", an interrupt is triggered. Clearing the HIRQn:DIRQ flag clears the interrupt. When the USB bus is reset, it determines that the device is disconnected and sets the HSTATEn:CSTAT bit to "0", but the HIRQn:DIRQ bit is not set to "1".

62.6. Token Flow Chart

The flow chart of each USB Mini-host token.

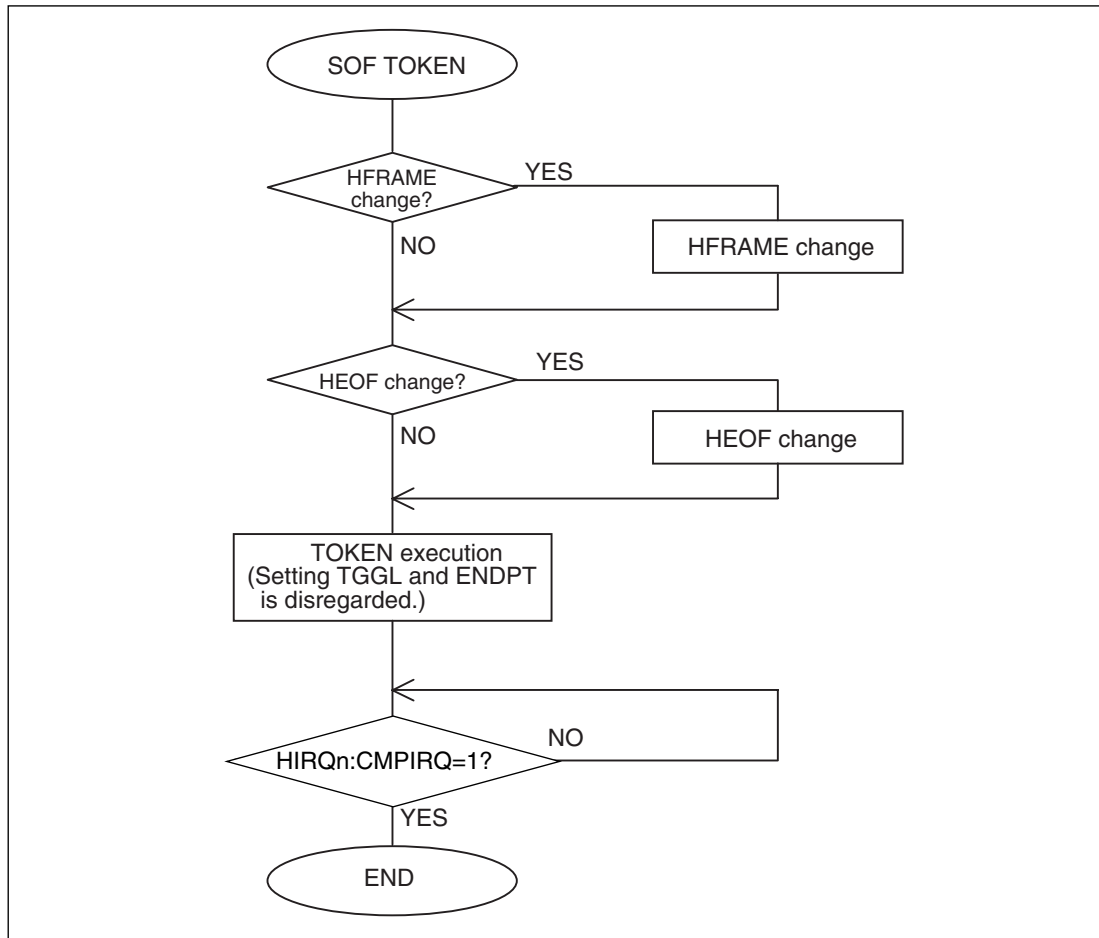
■ IN, OUT, SETUP Token

Figure 62.6-1 Flow Chart at IN, OUT, SETUP Token



■ SOF Token

Figure 62.6-2 Flow Chart of SOF Token



Chapter 63 USB Function

This chapter explains the functions and operation of the USB Function.

63.1. USB Function Overview

The USB Function is an interface that supports the USB (Universal Serial Bus) 1.1 communication protocol. Only Full speed transfer (12 Mbps) is supported.

■ Features of USB Function

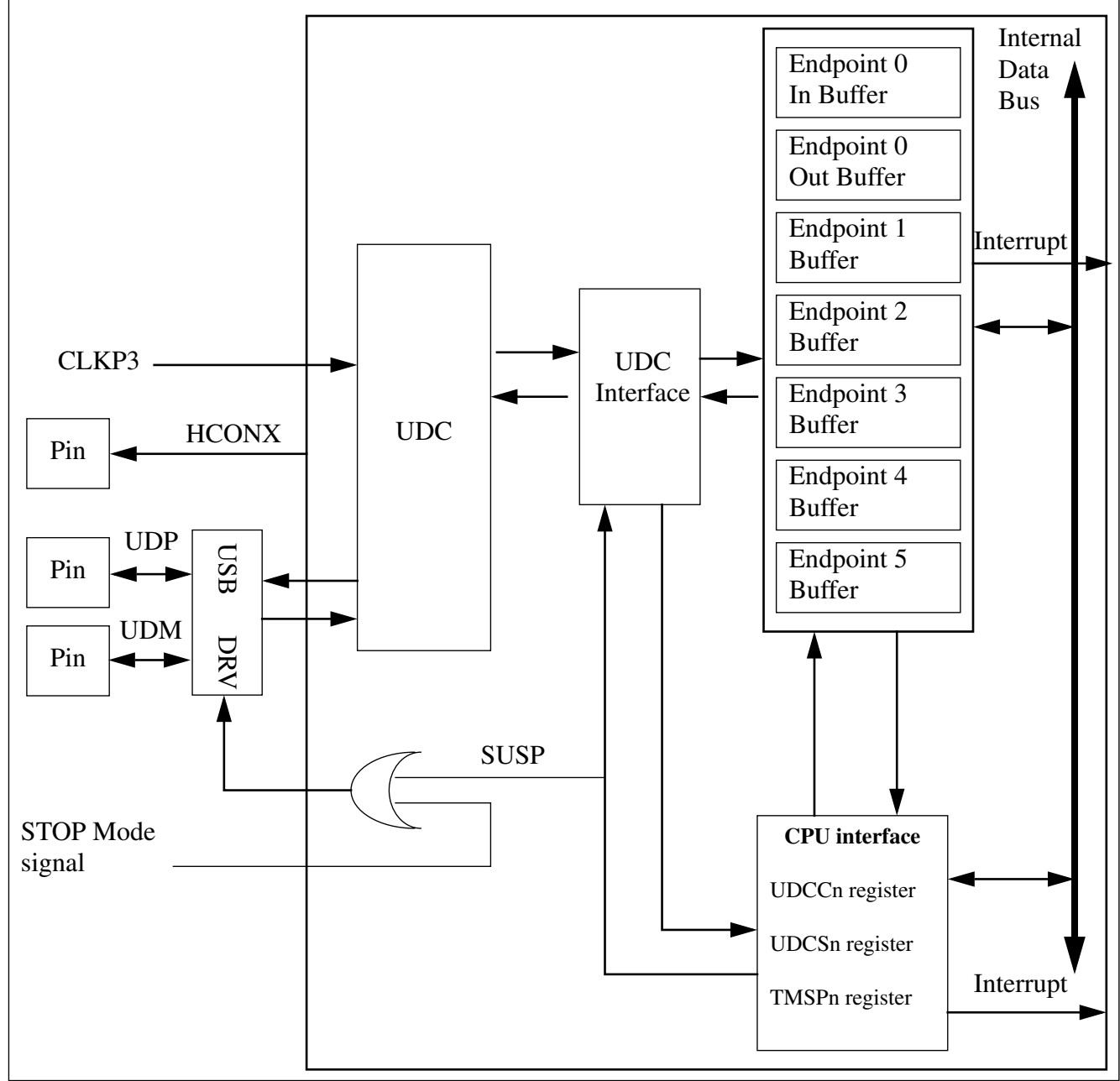
- Full speed (12 Mbps) is supported, corresponds to USB Full Speed.
- Automatically responds to the Device Status commands..
- Bit stripping, bit stuffing and automatic generation and check of CRC5 and CRC16.
- Toggle check by data synchronization bit.
- Automatic response to all standard commands except GetDescriptor, SetDescriptor and SynchFrame commands. These commands can be processed the same way as the class vendor commands.
- The class vendor commands can be received as data and handled by firmware.
- Supports up to six endpoints (endpoint 0 is fixed for control transfers).
- Two built-in transfer data buffers for each endpoint. Each of the two built-in buffers dedicated to IN and OUT, respectively for endpoint 0.
- Supports automatic transfer mode for DMA data transfer (except buffers for endpoint 0).

63.2. USB Function Block Diagram

Description of the USB Function.

■ Block Diagram of USB Function

Figure 63.2-1 Block Diagram of USB Function



63.3. USB Function Register Description

Description of the USB function registers.

■ Register List of the USB function

Table 63.3-1 Register List

15	8	7	0
UDCCn			(R/W)
EP0Cn			(R/W)
EP1Cn			(R/W)
EP2Cn			(R/W)
EP3Cn			(R/W)
EP4Cn			(R/W)
EP5Cn			(R/W)
TMSPn			(R)
UDCIEn		UDCSn	(R/W)
EP0ISn			(R/W)
EP0OSn			(R/W)
EP1Sn			(R/W)
EP2Sn			(R/W)
EP3Sn			(R/W)
EP4Sn			(R/W)
EP5Sn			(R/W)
EP0DTn			(R/W)
EP1DTn			(R/W)
EP2DTn			(R/W)
EP3DTn			(R/W)
EP4DTn			(R/W)
EP5DTn			(R/W)
8bits			8 bits

Table 63.3-2 Registers of USB function

bit	7	6	5	4	3	2	1	0	
	RST	RESUM	HCONX	USTP	Reserved	Reserved	RFBK	PWC	UDC control register (UDCCn)
bit	15	14	13	12	11	10	9	8	
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
bit	7	6	5	4	3	2	1	0	
	Reserved	PKS0							EP0 control register (EP0Cn)

Table 63.3-2 Registers of USB function

bit	15	14	13	12	11	10	9	8	
	-	-	-	-	Reserved	Reserved	STAL	Reserved	
bit	7	6	5	4	3	2	1	0	
	PKS 1								EP1 control register (EP1Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	PKS1	
bit	7	6	5	4	3	2	1	0	
	Reserved	PKS 2							EP2 control register (EP2Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved	
bit	7	6	5	4	3	2	1	0	
	Reserved	PKS 3							EP3 control register (EP3Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved	
bit	7	6	5	4	3	2	1	0	
	Reserved	PKS 4							EP4 control register (EP4Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved	
bit	7	6	5	4	3	2	1	0	
	Reserved	PKS 5							EP5 control register (EP5Cn)
bit	15	14	13	12	11	10	9	8	
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved	
bit	7	6	5	4	3	2	1	0	
	TMSP								Time stamp register (TMSPn)
bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	TMSP			
bit	7	6	5	4	3	2	1	0	
	-	-	SUSP	SOF	BRST	WKUP	SETP	CONF	UDC status register (UDCSn)
bit	15	14	13	12	11	10	9	8	
	Reserved	Reserved	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE	UDC interruption permission register (UDCIEn)
bit	7	6	5	4	3	2	1	0	

Table 63.3-2 Registers of USB function

-								EP0I status register (EP0ISn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	-	-	-	DRQI	-	-
bit	7	6	5	4	3	2	1	0
	Reserved	SIZE						EP0O status register (EP0OSn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQOIE	SPKIE	-	-	DRQO	SPK	Reserved
bit	7	6	5	4	3	2	1	0
	SIZE							EP1 status register (EP1Sn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	SIZE
bit	7	6	5	4	3	2	1	0
	Reserved	SIZE						EP2 status register (EP2Sn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
bit	7	6	5	4	3	2	1	0
	Reserved	SIZE						EP3 status register (EP3Sn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
bit	7	6	5	4	3	2	1	0
	Reserved	SIZE						EP4 status register (EP4Sn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
bit	7	6	5	4	3	2	1	0
	Reserved	SIZE						EP5 status register (EP5Sn)
bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
bit	7	6	5	4	3	2	1	0
	BFDT							EP0 data register (EP0DTn)
bit	15	14	13	12	11	10	9	8
	BFDT							
bit	7	6	5	4	3	2	1	0

Table 63.3-2 Registers of USB function

	BFDt								EP 1 data register (EP1DTn)
bit	15	14	13	12	11	10	9	8	
	BFDt								
bit	7	6	5	4	3	2	1	0	
	BFDt								EP 2 data register (EP2DTn)
bit	15	14	13	12	11	10	9	8	
	BFDt								
bit	7	6	5	4	3	2	1	0	
	BFDt								EP 3 data register (EP3DTn)
bit	15	14	13	12	11	10	9	8	
	BFDt								
bit	7	6	5	4	3	2	1	0	
	BFDt								EP 4 data register (EP4DTn)
bit	15	14	13	12	11	10	9	8	
	BFDt								
bit	7	6	5	4	3	2	1	0	
	BFDt								EP 5 data register (EP5DTn)
bit	15	14	13	12	11	10	9	8	
	BFDt								

MB91460 Series

63.3.1 UDC Control Register (UDCCn)

UDC control register (UDCCn) controls the UDC (USB Device Controller Core) circuit.

■ UDC Control Register (UDCCn)

Figure 63.3-1 UDC Control Register (UDCCn)

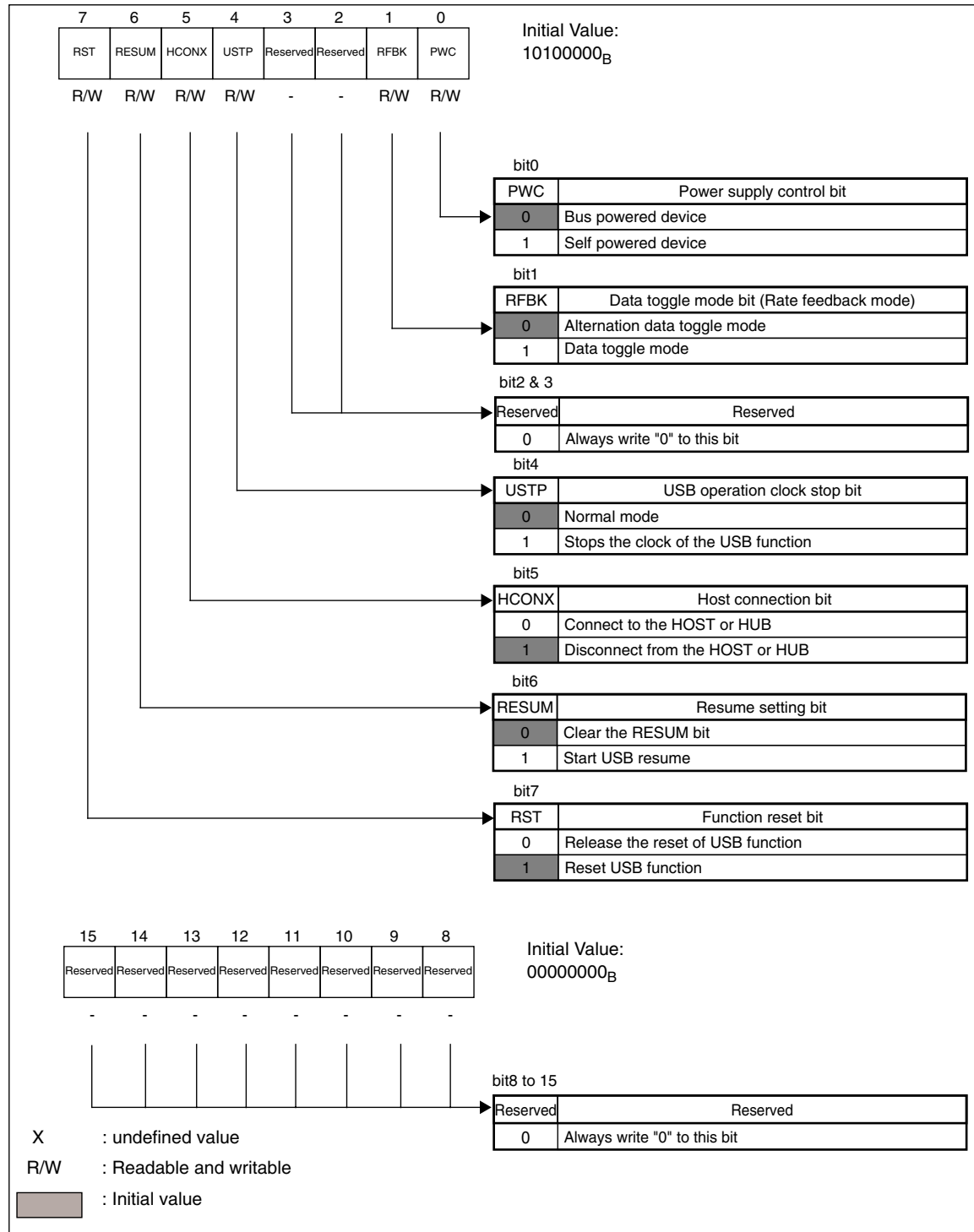


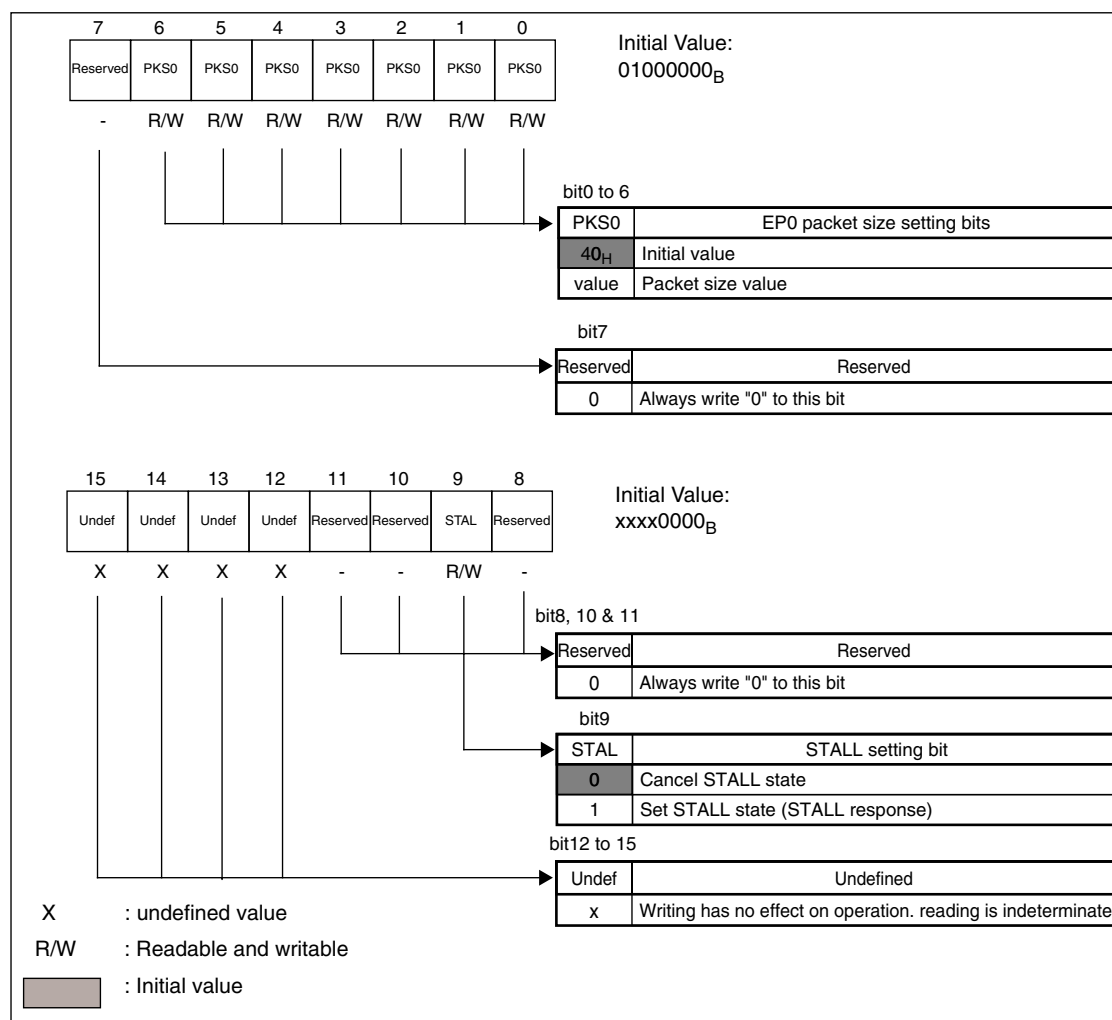
Table 63.3-3 Functional Description of each bit of the UDC control register (UDCC).

	Bit names	Function
Bit7	RST	Reset the whole USB function. Apply a reset to the USB Function by setting the RST bit to "1" when connecting a cable to the HOST. As the initial value of this bit is "1" i.e the USB function is reset, write "0" to this bit to release the USB function from its reset status. RST bit initializes the corresponding bits of the timestamp register, UDC status register, and interrupt enable register at once. In addition, since it initialises BFINI bit of EP0I, EP0O, and EP1 to EP5 status registers at the same time, first clear the RST bit (which does not clear the BFINI bits) after initialization and then clear the BFINI bit of the endpoint used.
Bit6	RESUM	When the USB function is in remote Wake-up enabled status (or DEVICE_REMOTE_WAKEUP feature is set with the SET_FEATURE command by the host) and is in suspend state, the resume operation is started by writing 1 to the RESUM bit. For resume of the USB function, please set the RESUM bit to "1" and then clear the RESUM bit to "0".
Bit5	HCONX	Controls a switch between an external pull-up resistor and the USB data line to make the HOST or HUB recognize that USB function is connected. Even if the external pull-up resistor is ON and the Host or Hub recognizes that USB function is connected, USB Function module ignores the bus reset and the command on the USB bus when the HCONX bit is set to "1".
Bit4	USTP	Stops the clock to the USB function. Setting the USTP bit to "1" stops the clock to the USB module to reduce power consumption. When you set USTP bit to "1" but are not going to select STOP mode after it, at first set RST bit to "1" and wait for 3 cycles to make sure that the USB module is reset, and then set USTP bit to "1". Clearing the USTP bit and the RST bit at once is OK.
Bit3	Reserved	Please write "0". The bit always reads "0".
Bit2	Reserved	Please write "0". The bit always reads "0".
Bit1	RFBK	This bits selects the Data Toggle mode. If set to "0" it toggles the data PID between DATA0 and DATA1 only when the transfer has been successfully completed. If set to "1" it toggles the data PID between DATA0 and DATA1 unconditionally.
Bit0	PWC	This bit specifies the power supply mode (self or bus powered) for the USB function. (The setting of the PWC bit is reflected in the standard usb command: GetStatus command).

Note: Bits of the UDC control register (UDCCn) must not be changed when RST = '1' except RESUM and USTP bit. Do not change them while the USB is operating. Set and reset RESUM bit only when USB is in suspend mode and Remote Wake-up is allowed (DEVICE_REMOTE_WAKEUP bit is set by the SET_FEATURE command from the Host). When stop mode is selected, set USTP bit before entering the stop mode. After the stop mode is cancelled, clear the SUSP bit to "0" and then clear the USTP bit to "0".

MB91460 Series**63.3.2 EP0 Control Register (EP0Cn)**

EP0 control register (EP0Cn) is used to control the operation of endpoint 0.

■ EP0 Control Register (EP0Cn)**Figure 63.3-2 EP0 Control Register (EP0Cn)**

Note: Ensure that you must set the EP0 control register (EP0Cn), except bit 9 STAL, when both RST of the UDC control register (UDCCn) and BFINI bit of the EP0I/EP0O status register (EP0ISn / EP0OSn) are "1". Changing EP0Cn is not allowed while the USB is operating.

Table 63.3-4 Functional Description of each bit of the EP0 Control Register (EP0Cn).

Bit names		Function
Bit15-12	Undefined	<ul style="list-style-type: none"> Writing has no effect on operation. Reading is indeterminate.
Bit11-10	Reserved	<ul style="list-style-type: none"> Always write "0". These bits always read "0".
Bit9	STAL	<ul style="list-style-type: none"> Setting the STAL bit to "1" can put endpoint 0 in STALL state (STALL response). The STALL response is sent to the host till the STAL bit is set to "1". The USB Function returns from STALL state when it receives a normal SETUP packet after the STAL bit is cleared by writing "0" to it.
Bit8-7	Reserved	<ul style="list-style-type: none"> Always write "0". These bits always read "0".

Bit names		Function
Bit6-0	PKS0	<ul style="list-style-type: none">• It specifies the maximum number of bytes that can be transferred per packet. The maximum number of transfer bytes per packet for EndPoint0 is 64 bytes, which is a common setting for both IN and OUT packet transfers. Example: "08_H"Æ8 Byte, "40_H"Æ64 Byte (maximum specified value)• Value should always be greater than "00_H" and less than or equal to the maximum number of transfer bytes (40_H).

MB91460 Series

63.3.3 EP1 to EP5 Control Register (EP1Cn to EP5Cn)

EP1 to EP5 Control Register (EP1Cn to EP5Cn) are used to control the operation of endpoints 1 to 5.

■ EP1 to EP5 Control Register (EP1Cn to EP5Cn)

Figure 63.3-3 EP1 Control Register (EP1Cn)

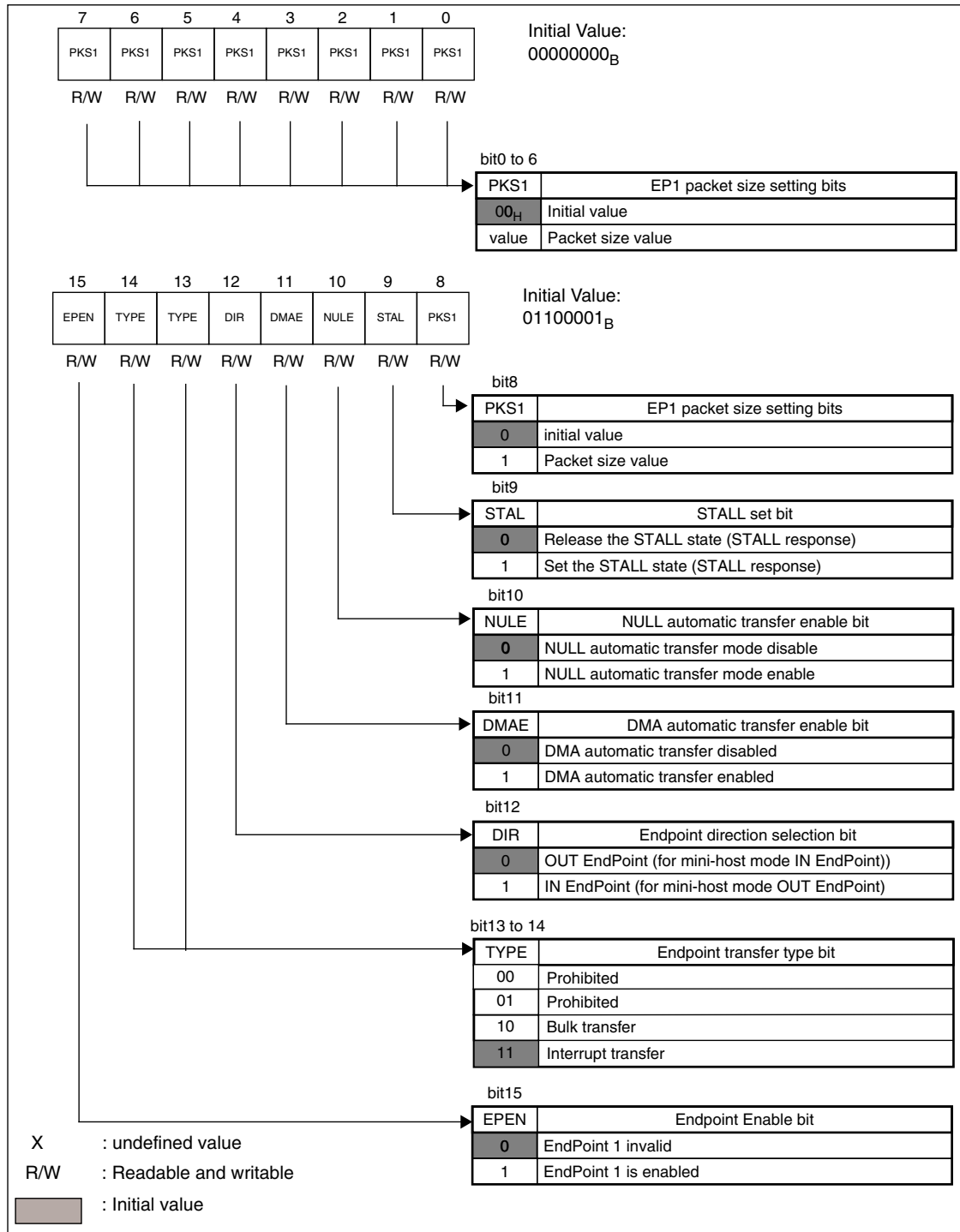
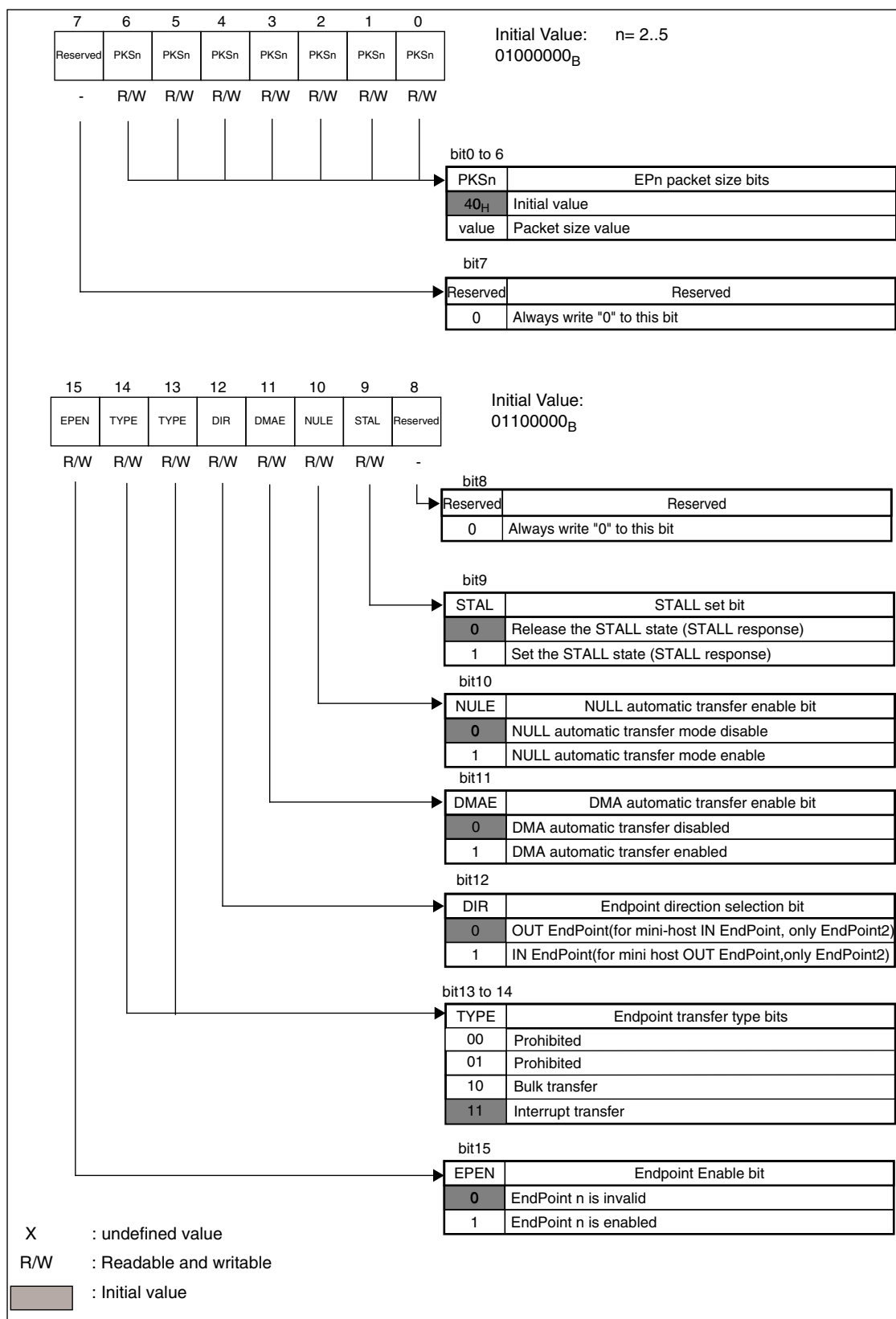


Figure 63.3-4 EP2 to EP5 Control register (EP2Cn to EP5Cn)



Note: EP1 to EP5 control registers (EP1Cn to EP5Cn) except DMAE, NULE, and STAL bits must be configured only when both UDCCn:RST bit and EPxSn:BFINI are "1". Do not write to these registers while the USB is operating.

Table 63.3-5 Functional Description of each bit of the EP1 to EP5 control register (EP1Cn to EP5Cn).

Bit names		Function									
Bit15	EPEN	<ul style="list-style-type: none"> Endpoint enable. Setting the EPEN bit allows an endpoint to be configured by the host as an end point for use in the USB Function. When this bit is set to "1", TYPE, DIR, and PKS of the EP1 to EP5 control registers become valid for configuration. 									
Bit14-13	TYPE	<ul style="list-style-type: none"> These two bits specify the transfer type for the corresponding endpoint, i.e Bulk or Interrupt transfer. 									
Bit12	DIR	<ul style="list-style-type: none"> This bit specifies the transfer direction of the endpoint For USB function, writing "1" to this bit configures the corresponding End Point as IN endpoint. Writing "0" configures the endpoint as OUT endpoint. In USB Host mode, only End Point 1 and 2 are valid. Writing "0" to this bit in host mode configures the corresponding End Point as IN endpoint and writing "1" to this bit configures the End Point as OUT endpoint. 									
Bit11	DMAE	<ul style="list-style-type: none"> DMA automatic transfer mode enable bit. If enabled, the DMA handles automatically the transmission and reception of data in sync with IN and OUT data requests from the HOST until the data bytes equal to the number set in DCT register of DMA is transferred. See "63.4.5 DMA Transfer Function" for details. Access to transmit and receive buffers by CPU is forbidden while the DMAE bit is set. Set the number of DMA transfer data to a multiple of the value set in the PKS bits of EP1 to EP5 control registers (EP1Cn to EP5Cn) when the endpoint is used for OUT transfer. 									
Bit10	NULE	<ul style="list-style-type: none"> Enable NULL transfer. See "63.4.6 NULL Transfer Function" for details. During IN transfer, the last packet transfer will be detected and a 0-byte data packet will be automatically sent when IN data transfer request arrives from the host and the automatic buffer transfer mode is set (DMAE=1). NULE bit has no effect on communications when transferring data to OUT direction or when the automatic buffer transfer mode is disabled. 									
Bit9	STAL	<ul style="list-style-type: none"> Set the endpoint in STALL status (STALL response). When the STAL bit is set, the USB function will respond to any requests from the HOST with a STALL handshake packet. The USB Function can return from STALL status after receiving a ClearFeature command from the host after the STAL bit is cleared by writing "0" to this bit. 									
Bit8	Reserved	<ul style="list-style-type: none"> Reserved for EP2 to EP5 Always write "0" to this bit. 									
Bit7	Reserved	<ul style="list-style-type: none"> Reserved for EP2 to EP5 Always write "0" to this bit. 									
Bit0-8: EP1Cn Bit0-6: EP2Cn- EP5Cn	PKS	<ul style="list-style-type: none"> This bit specifies the maximum number of bytes that can be transferred per packet. The following table shows the maximum number of bytes per packet to be transferred which can be specified for each endpoint: <table border="1"> <thead> <tr> <th>Endpoint</th><th>Max number of transfer</th><th>Possible values</th></tr> </thead> <tbody> <tr> <td>1</td><td>256 bytes</td><td>001h to 100h</td></tr> <tr> <td>2 to 5</td><td>64 bytes</td><td>01h to 40h</td></tr> </tbody> </table> <ul style="list-style-type: none"> Value should always be greater than "00H" and less than or equal to the maximum number of transfer bytes (100H or 40H). "00" must be written in bits 7 and 8 for endpoints 2 to 5. When the Data number automatic transfer mode is selected (DMAE=1), setting value less than 3 in the corresponding end point is forbidden. 	Endpoint	Max number of transfer	Possible values	1	256 bytes	001h to 100h	2 to 5	64 bytes	01h to 40h
Endpoint	Max number of transfer	Possible values									
1	256 bytes	001h to 100h									
2 to 5	64 bytes	01h to 40h									

63.3.4 Time Stamp Register (TMSPn)

The time stamp register (TMSPn) displays a frame number when a SOF packet is received.

■ Time Stamp register (TMSPn)

Figure 63.3-5 Time Stamp register (TMSPn)

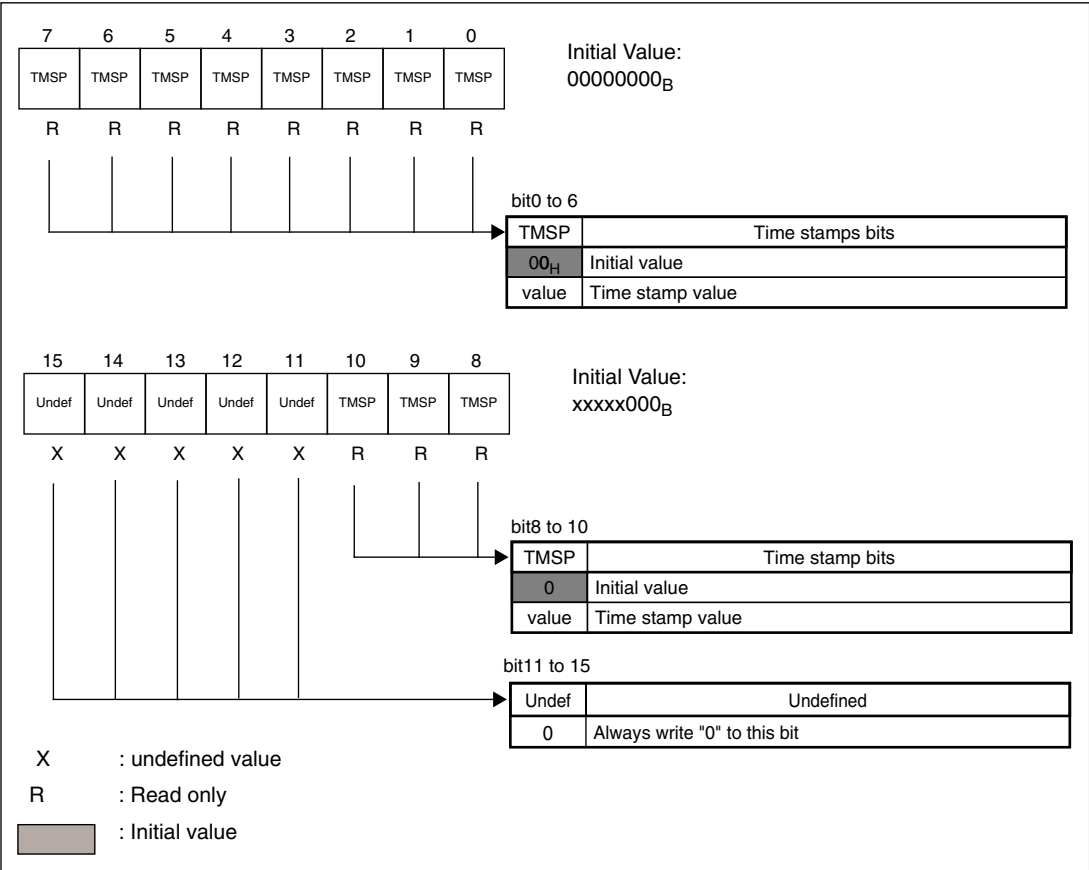


Table 63.3-6 Functional Description of each bit of the time stamp register (TMSPn).

Bit names		Function
Bit15-11	Undefined	<ul style="list-style-type: none">No functionality
Bit10-0	TMSP	<ul style="list-style-type: none">It contains the Frame number of the current SOF packet. These bits are used when a SOF packet is received.

MB91460 Series

63.3.5 UDC Status register (UDCSn)

The UDC Status Register (UDCSn) indicates the status of the USB communication. Each bit in the register, except SETP indicates an interrupt factor and raises an interrupt to the CPU if the corresponding interrupt enable bit in the UDC Interruption Enable register (UDCIEn) is set.

■ UDC Status Register (UDCSn)

Figure 63.3-6 UDC Status register

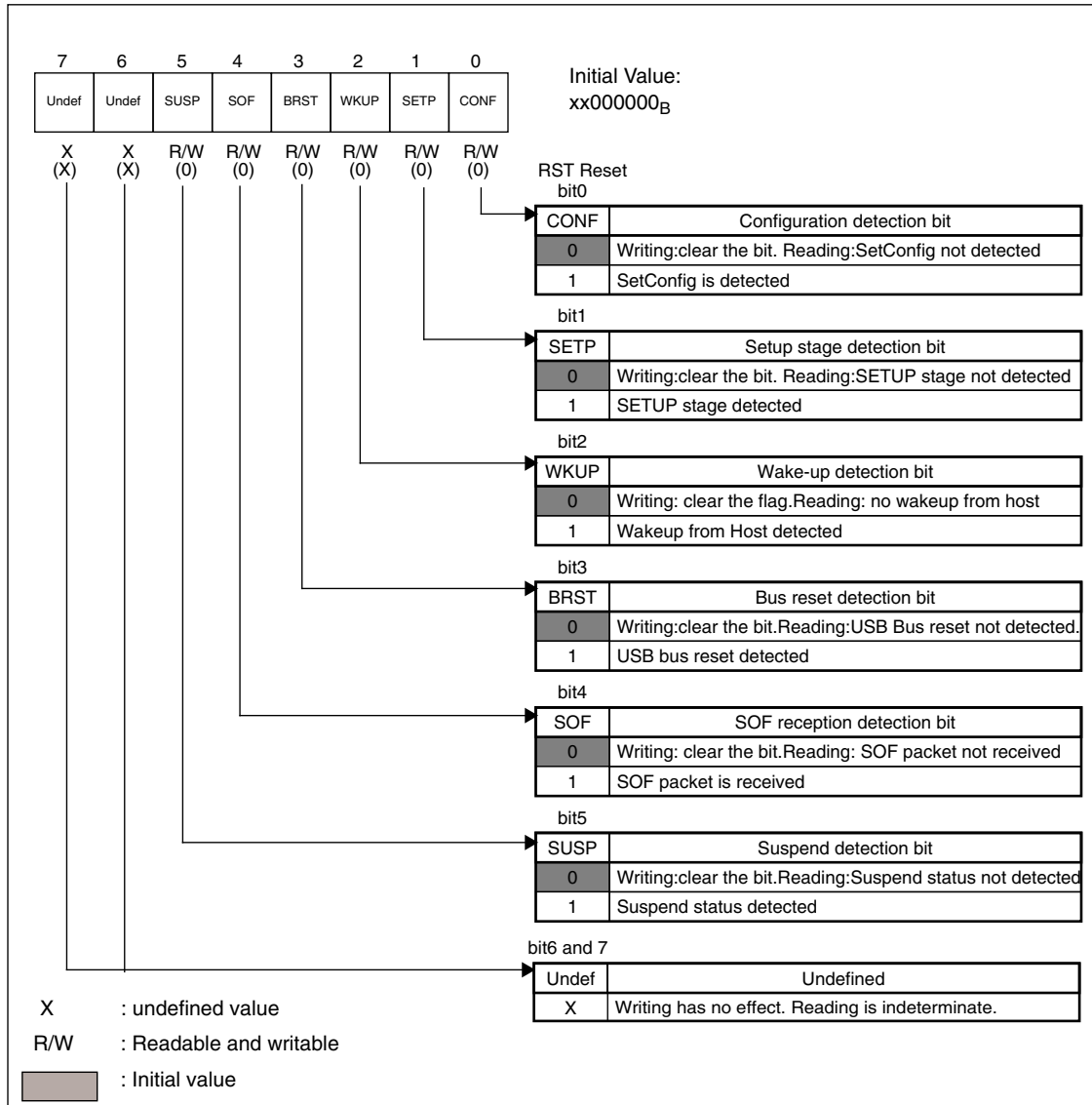


Table 63.3-7 Functional Description of each bit of the UDC status register (UDCSn):

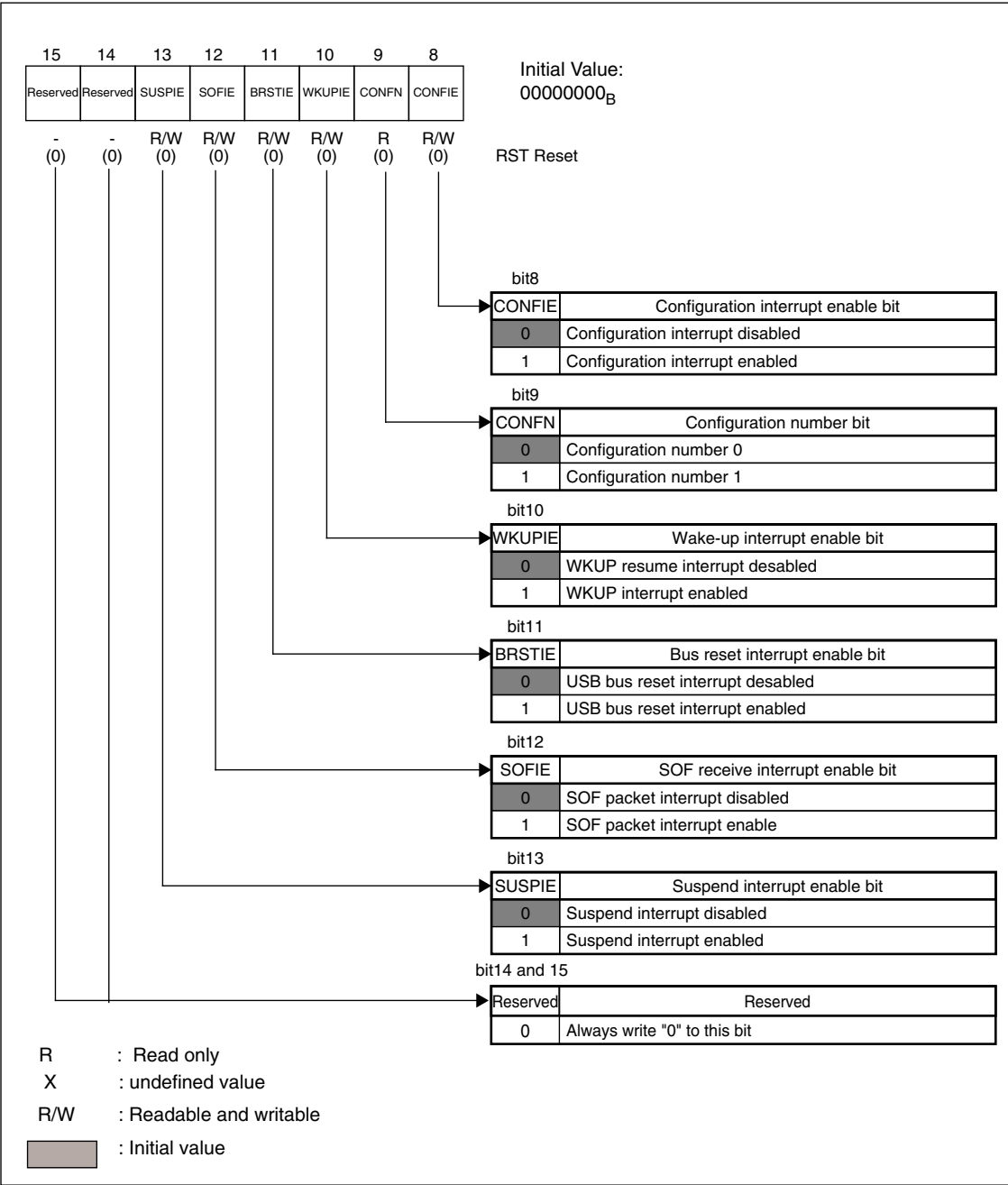
Bit names		Function
Bit7-6	Undefined	<ul style="list-style-type: none"> No functionality
Bit5	SUSP	<ul style="list-style-type: none"> This flag indicates that the USB function is in suspend mode. SUSP bit triggers the USB_F10 interrupt. Writing "1" is ignored. Clear by writing "0" to the bit. "1" is read at the read modification write access.
Bit4	SOF	<ul style="list-style-type: none"> Flag indicating the reception of a SOF packet. The value of the time stamp register (TMSPn) is updated when SOF is set to "1". The SOF flag triggers the USB_F10 interrupt. Writing "1" is ignored. Clear by writing "0" to this bit. "1" is read at the read modification write access.
Bit3	BRST	<ul style="list-style-type: none"> This flag indicates the reset of USB bus. It triggers the USB_F10 interrupt. Set all USB registers again after initializing the USB function with UDCCn:RST when bus reset is detected (BRST = "1"). Writing "1" to this bit is ignored. Clear by writing "0". "1" is read at the read modification write access.
Bit2	WKUP	<ul style="list-style-type: none"> This flag indicates that the USB function has returned from the suspend mode. This flag triggers the USB_F10 interrupt. A remote wake-up (by setting the RESUM bit) or a wake-up request from the HOST causes the USB function to return from suspend state. WKUP bit is automatically set only by a resume request from the HOST. If the RESUM bit in UDCCn register is set to "1", WKUP flag is not set even if wakeup is caused by the host. Writing "1" is ignored. Clear by writing "0". "1" is read at the read modification write access.
Bit1	SETP	<ul style="list-style-type: none"> This bit indicates that the received data belongs to the SETUP stage of USB control transfer. This bit is not set when the USB function automatically responds to any standard commands. SETP bit does not trigger any interrupt. Writing "1" has no effect. Clear by writing "0". "1" is read at the read modification write access.
Bit0	CONF	<ul style="list-style-type: none"> It is set when the USB command SetConfig has been received and the USB Function is configured. The CONF bit trigger the USB_F10 interrupt. Writing "1" is ignored. Clear by writing "0". "1" is read at the read modification write access.

63.3.6 UDC Interrupt Enable Register (UDCIEn)

The UDC Interrupt Enable Register (UDCIEn) is a register that enables each flag in the UDC status register (UDCSn) to trigger an interrupt. (except the CONFN bit).

■ UDC Interrupt Enable register (UDCIEn)

Figure 63.3-7 UDC Interrupt Enable Register (UDCIEn)



The function of each bit in the UDC interrupt register (UDCIEn) is described in the following table.

Table 63.3-8 Functional Description of each bit of the UDC Interruption Enable Register (UDCIEn).

Bit names		Function
Bit15-14	Reserved	<ul style="list-style-type: none">Always write "0" to this bit. Read value of this bit is always "0"
Bit13	SUSPIE	<ul style="list-style-type: none">Enable interrupt caused by SUSP flag.
Bit12	SOFIE	<ul style="list-style-type: none">Enable interrupt caused by SOF flag.
Bit11	BRSTIE	<ul style="list-style-type: none">Enable interrupt caused by BRST flag.
Bit10	WKUPIE	<ul style="list-style-type: none">Enable interrupt caused by WKUP flag.
Bit9	CONFN	<ul style="list-style-type: none">This bit displays the configuration number.It is updated when CONF flag of UDCSn is set by SET_CONFIGURATION command from the host.
Bit8	CONFIE	<ul style="list-style-type: none">Enable interrupt caused by CONF flag of UDCSn.

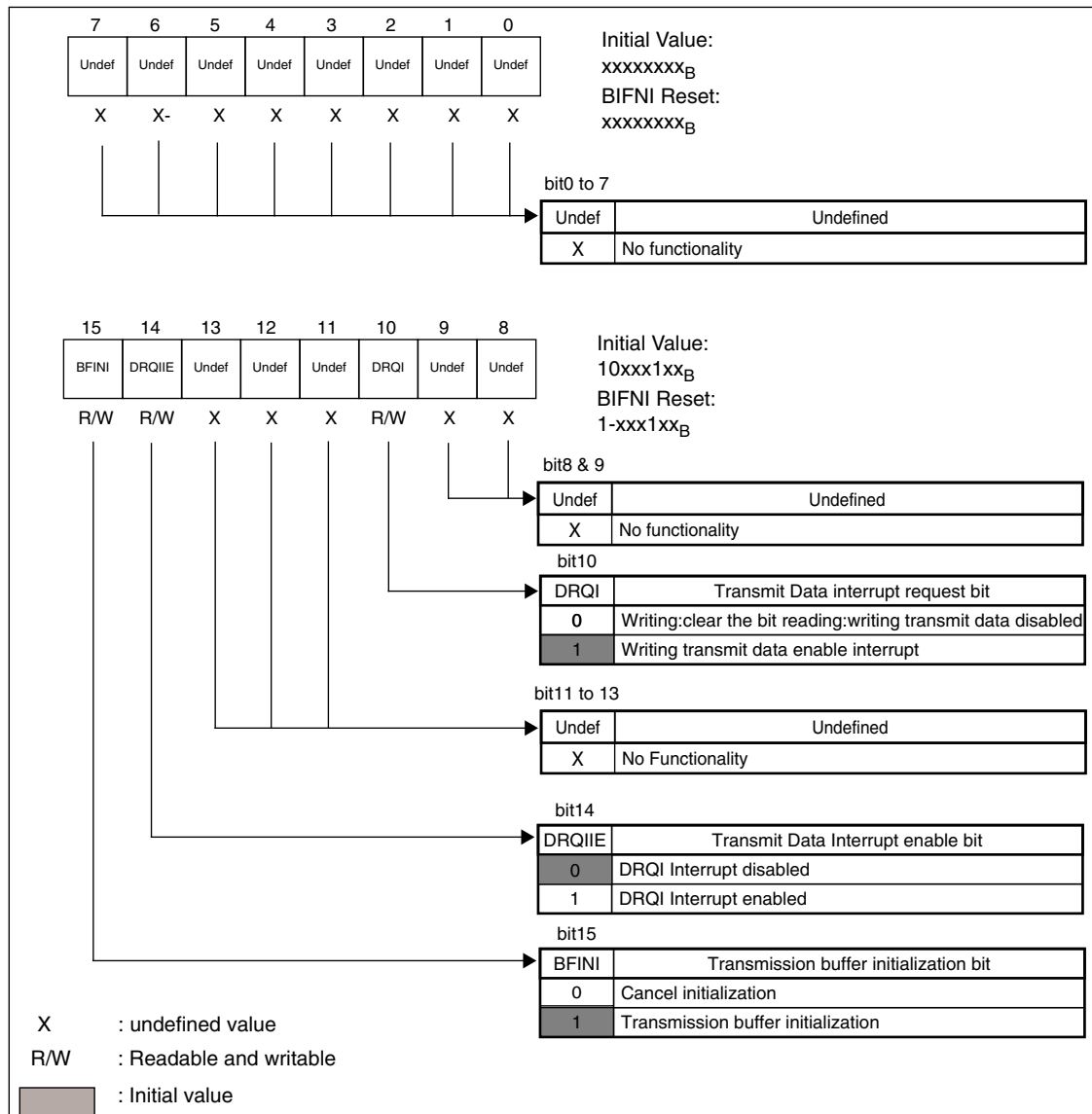
MB91460 Series

63.3.7 EP0I Status Register (EP0ISn)

The EP0I Status Register (EP0ISn) displays the status of IN direction transfer for EndPoint 0.

■ EP0I Status Register (EP0ISn)

Figure 63.3-8 EP0I Status register (EP0ISn)



The function of each bit in the EP0I status register (EP0ISn) is described in the following.

Table 63.3-9 Functional Description of each bit of the EP0I Status Register (EP0ISn).

Bit names		Function
Bit15	BFNI	<ul style="list-style-type: none"> This bit initializes the transmission buffer. The BFINI bit is automatically set when the RST flag in the UDC Control Register (UDCCn) is set to "0". When the reset operation has been performed with the UDCCn:RST bit, clear the UDCCn:RST bit before clearing the BFINI bit. The BFINI bit initializes a buffer and the DRQI bit. Before initializing the buffer, please ensure that the DRQI or DRQO bit is set to "1" that means there is no access from the HOST, and set the STAL bit (if it is necessary).
Bit14	DRQIIE	<ul style="list-style-type: none"> DRQI interrupt enable. It enables an interrupt to be triggered when DRQI flag is set.
Bit13-11	Undefined	<ul style="list-style-type: none"> No functionality.
Bit10	DRQI	<ul style="list-style-type: none"> This bit indicates that an IN packet has been successfully transferred from the EP0 to the host. The data has been read out from the transmission buffer, and the next data to be transmitted can be written into the buffer. After the data is written to the transmission buffer, the DRQI must be cleared. When DRQI = "0", writing "0" is prohibited. When DRQI = "1", writing data to the transmission buffer is enabled. Clearing the DRQI bit indicates that writing data to the transmit buffer is completed. If IN packet request is performed when DRQI = "1", NAK response is sent to the HOST automatically. So this bit is to be cleared to process IN transfer request. The DRQI bit triggers USB_EP0IN0 interrupt. Writing "1" to this bit is ignored. Clear by writing "0". "1" is read at the read modification write access.
Bit9-0	Undefined	<ul style="list-style-type: none"> No functionality

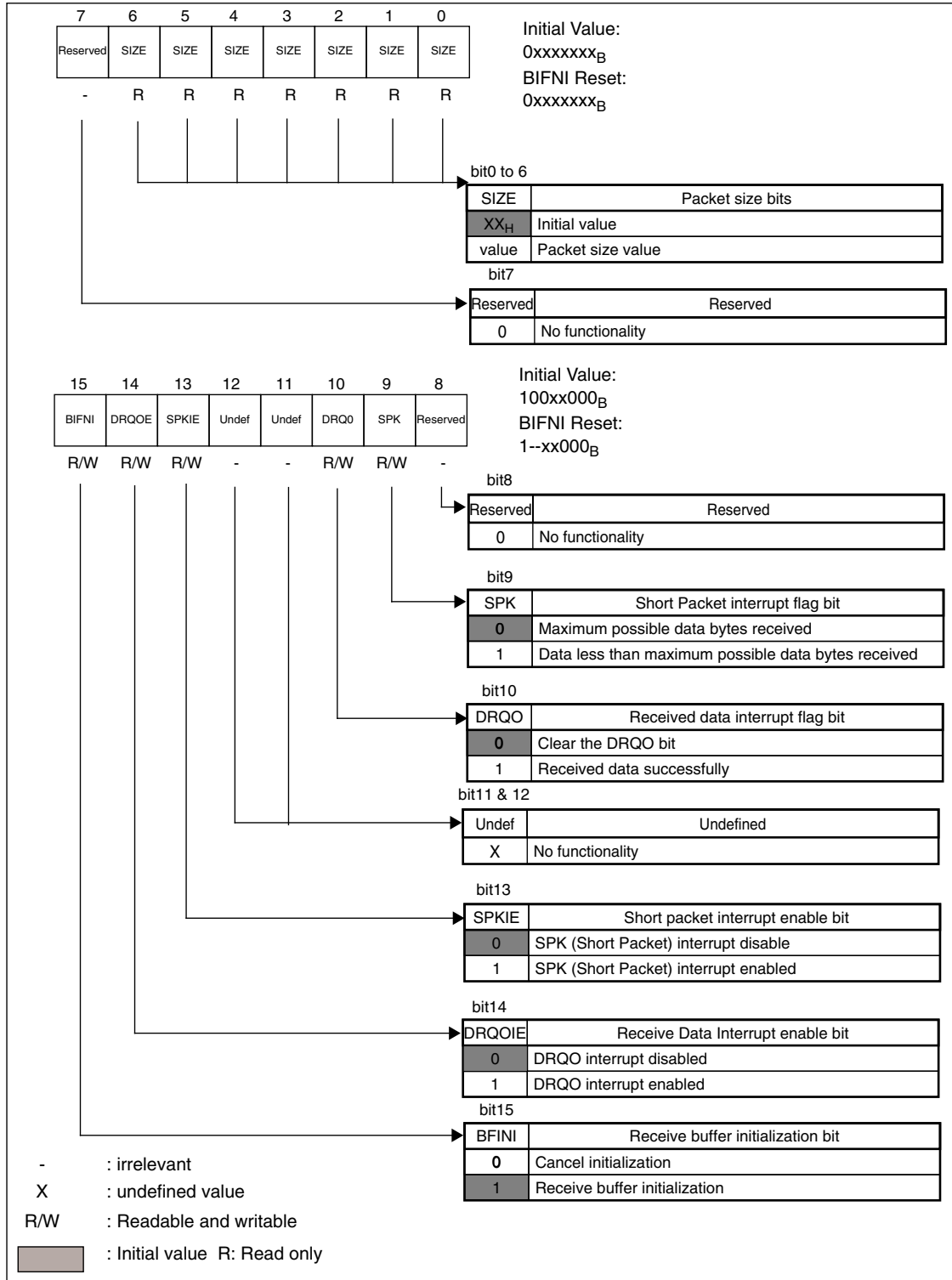
MB91460 Series

63.3.8 EP00 Status Register (EP0OSn)

The EP00 Status Register (EP0OSn) displays status of OUT direction transfer of the endpoint 0.

■ EP00 Status Register (EP0OSn)

Figure 63.3-9 EP00 Status Register (EP0OSn)



The function of each bit in the EP0O status register (EP0OSn) is described in the following.

Table 63.3-10 Functional Description of each bit of the EP0O Status Register (EP0OSn).

Bit names		Function
Bit15	BFINI	<ul style="list-style-type: none"> This bit initializes the receive buffer. The BFINI bit is automatically set when the RST bit in the UDC control register (UDCCn) is set to "1". When the reset operation has been performed with the UDCCn:RST bit, clear the UDCCn:RST bit before clearing the BFINI bit. The BFINI bit initializes a buffer and the DRQO bit and SPK bit. Before initializing the buffer please ensure that the DRQI or DRQO bit is set to "1" that means there is no access from the HOST, and set the STAL bit (if it is necessary).
Bit14	DRQOIE	<ul style="list-style-type: none"> DRQO interrupt enable. It enables an interrupt to be triggered when DRQO flag is set.
Bit13	SPKIE	<ul style="list-style-type: none"> This bit enables interrupt to be triggered when SPK flag is set.
Bit12-11	Undefined	<ul style="list-style-type: none"> No functionality.
Bit10	DRQO	<ul style="list-style-type: none"> It indicates that an OUT packet has been successfully received from the host, data has been written into the receive buffer and the received data can be read from the receive buffer. After the receive buffer is read, DRQO must be cleared. When DRQO = "0", writing "0" is prohibited. When DRQO = "1", receive buffer is not updated. The buffer is enabled to be updated when DRQO is cleared. When OUT packet request is performed by the host when DRQO bit set to "1", NAK response is sent to the HOST automatically. The DRQO bit trigger an interrupt USB_EP0OUT0. Writing "1" is ignored. Clear by writing "0". "1" is read at the read modification write access.
Bit9	SPK	<ul style="list-style-type: none"> This bit indicates that the number of bytes of the data packet successfully received from the HOST is less than the maximum packet size value set in EP0Cn:PKS. (zero length packet will also set this bit). This bit triggers an interrupt USB_F20. Writing "1" is ignored. Clear by writing "0". "1" is read at the read modification write access.
Bit8-7	Undefined	<ul style="list-style-type: none"> No functionality
Bit6-0	SIZE	<ul style="list-style-type: none"> When OUT packets have been transferred from host to EP0, the number of data bytes that have been written into the reception buffer is displayed. The SIZE bits are updated to a valid value when EP0OSn:DRQO flag is set. Example: 8 bytes Æ "08_H", 64 bytes Æ "40_H" (maximum value)

MB91460 Series

63.3.9 EP1 to EP5 Status Register (EP1Sn to EP5Sn)

The EP1 to EP5 Status Registers (EP1Sn to EP5Sn) display status of endpoint 1 to endpoint 5.

■ EP1 to EP5 Status Register (EP1Sn to EP5Sn)

Figure 63.3-10 EP1 Status Register (EP1Sn)

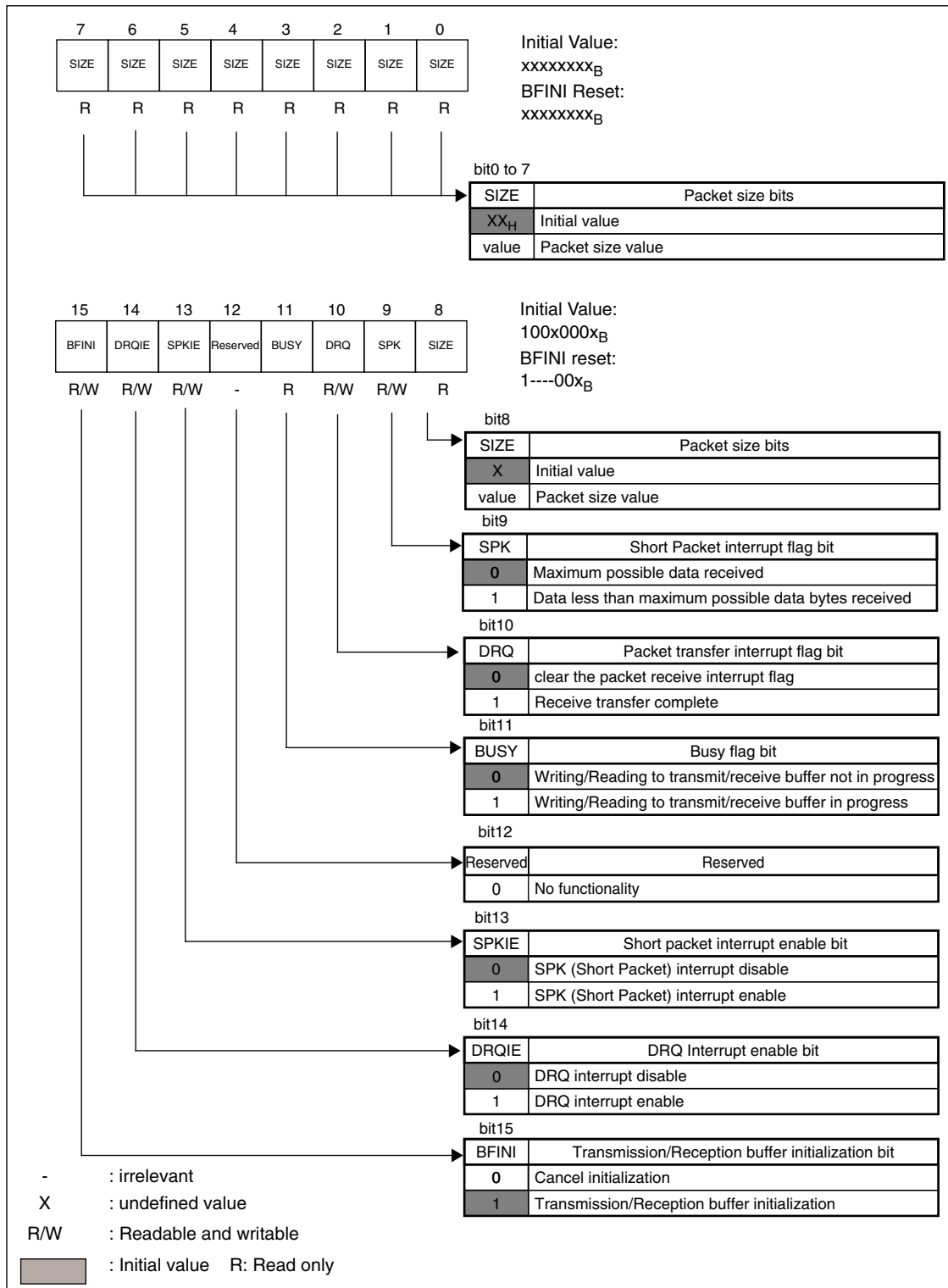
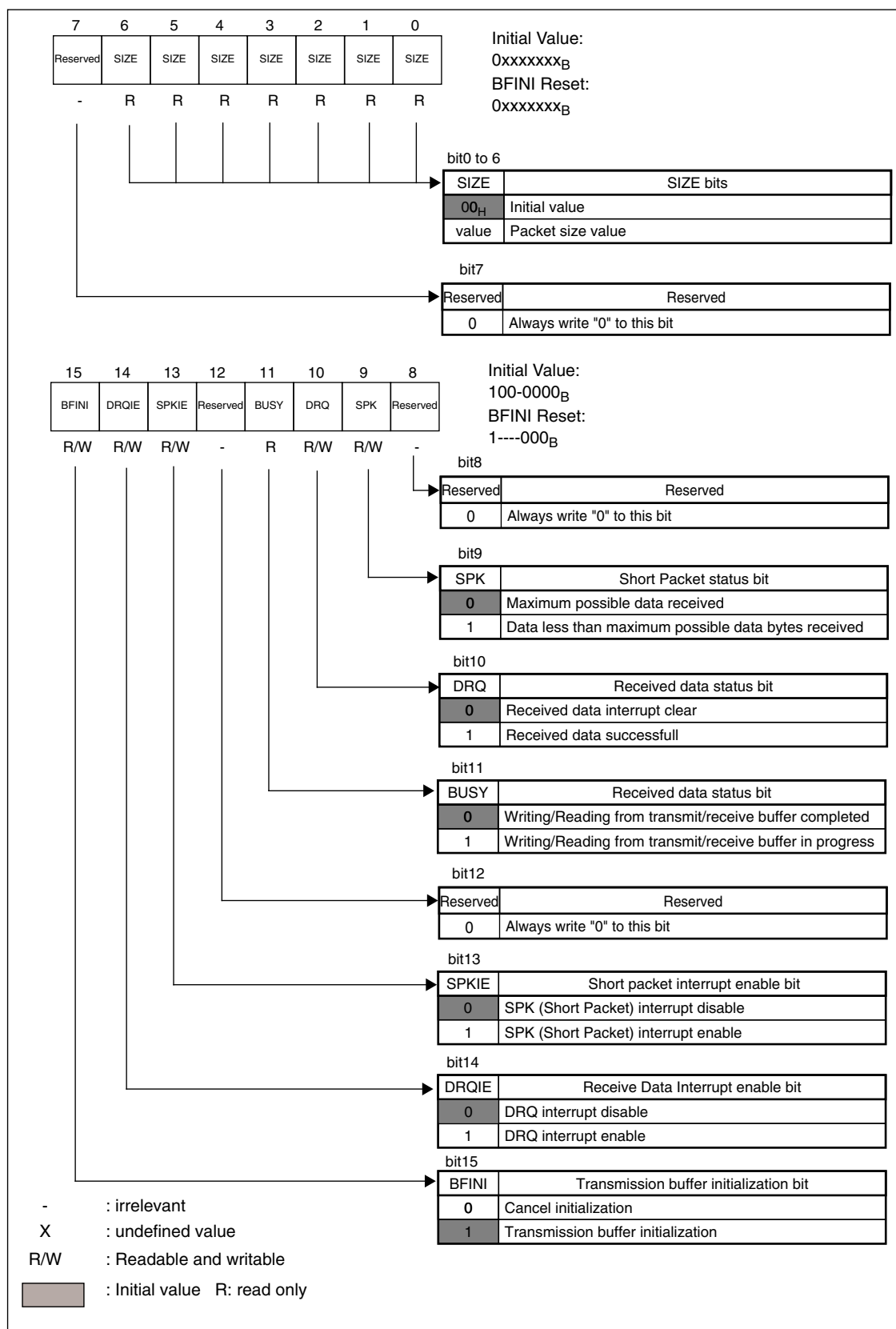


Figure 63.3-11 EP2 to EP5 Status Register (EP2Sn to EP5Sn)



The function of each bit in the EP1 to EP5 status register (EP1Sn to EP5Sn) is described in the following.

Table 63.3-11 Functional Description of each bit of the EP1 to EP5 Status Register (EP1Sn to EP5Sn).

Bit names		Function
Bit15	BFINI	<p>This bit initializes the transmission/reception buffer.</p> <p>The BFINI bit is automatically set when the RST flag in the UDC control register (UDCCn) is set to 1.</p> <p>When the reset operation has been performed with the RST bit, clear the RST bit before clearing the BFINI bit.</p> <p>The transmission/receive buffer for EP1 to EP5 has a configuration of double buffers. Initialization by BFINI bit initializes the double buffers, the DRQ and SPK bits all at once.</p> <p>Before initializing the buffer please ensure that the DRQ bit is set to "1", and ensure that BUSY="0" that means there is no access from the HOST, and set the STAL bit.</p>
Bit14	DRQIE	<p>DRQ interrupt enable. It enables an interrupt to be triggered when the DRQ flag of one of the EP1Sn to EP5Sn is set.</p> <p>In automatic buffer transfer mode (DMAE = "1"), DMA settings must be enabled before setting DRQIE.</p>
Bit13	SPKIE	This bit enables an interrupt to be triggered by the SPK flag of each EP1Sn to EP5Sn register.
Bit12	Undefined	No functionality.
Bit11	BUSY	<p>This bit indicates that writing into the transmission/receive buffer or reading it for sending data to/from the host is in progress.</p> <p>It is set/reset automatically.</p> <p>When the DRQ bit is set and also the BUSY flag is set, it means that the host is accessing one of the double buffers that is different from the buffer accessed by the CPU or DMA.</p> <p>Normally there is no need to refer to the BUSY bit when controlling the buffers. But before initializing the buffer, please ensure that the DRQ bit is set to "1", and ensure that the BUSY="0" that means there is no access from the HOST, and set the STAL bit.</p>
Bit10	DRQ	<p>This bit indicates that packet transfer for EPx has been successfully done and data processing is needed.</p> <p>The EPxSn:DRQ bit triggers corresponding USB_EPx0 interrupt .</p> <p>Writing "1" to this bit is ignored.</p> <p>Clear by writing "0". "1" is read at the read modification write access.</p> <p>After the data to be sent to the host is written to the transmission buffer (in response to the IN packet request from the host), the DRQ bit must be cleared. When the OUT packet transfer is completed DRQ bit is automatically set to "1". It should be cleared when the data from the receive buffer is completely read by the CPU/DMA.</p> <p>When DRQ = "0", writing "0" is prohibited.</p> <p>When the automatic buffer transfer mode (DMAE=1) is not used after the data read or write of transmission and reception buffers is completed, "0" must be written to the DRQ bit. When DRQ bit is cleared, access buffer is switched. If the transfer direction is set to IN direction and the DRQ bit is "1" and the DRQ bit is cleared without writing data to the transmission buffer, 0-byte data packet is sent to the host.</p> <p>In the initial setting, when the DIR of the EP1 to EP5 control register (EP1Cn to EP5Cn) is set to "1", DRQ bit of the corresponding end point is set at the same time.</p>
Bit9	SPK	<p>This bit indicates that the number of data bytes received successfully from the HOST is less than the maximum packet size value set in EPxCn:PKS (including 0 byte data packets).</p> <p>The SPK bit triggers USB_F20 interrupt.</p> <p>Writing "1" to this bit is ignored.</p> <p>Clear by writing "0". "1" is read at the read modification write access.</p> <p>The SPK bit is not set during IN data transfer.</p>

Bit names		Function									
Bit8-7	Reserved/ SIZE (EP1Sn)	Do not use for EP2Sn to EP5Sn. For EP1Sn these are the SIZE bits (see below)									
Bit8-0: EP1Sn Bit6-0: EP2Sn - EP5Sn	SIZE	<p>When OUT packets have been transferred to EPx, the number of data bytes that has been written into the reception buffer is displayed by SIZE bits. The SIZE bit is updated to a valid value when EPxSn:DRQ flag is set. The following table displays the maximum transferable number of data for each endpoint 1 to 5:</p> <table border="1"> <thead> <tr> <th>Endpoint</th><th>Max number of transfer</th><th>Possible values</th></tr> </thead> <tbody> <tr> <td>1</td><td>256 bytes</td><td>000h to 100h</td></tr> <tr> <td>2 to 5</td><td>64 bytes</td><td>00h to 40h</td></tr> </tbody> </table> <p>Since SIZE displays data packet size written in the buffer by the host during OUT packet transfer, any SIZE value read when an IN direction transfer is performed, is meaningless.</p>	Endpoint	Max number of transfer	Possible values	1	256 bytes	000h to 100h	2 to 5	64 bytes	00h to 40h
Endpoint	Max number of transfer	Possible values									
1	256 bytes	000h to 100h									
2 to 5	64 bytes	00h to 40h									

63.3.10 EP0 to EP5 Data Register (EP0DTn to EP5DTn)

The EP0 to EP5 Data Registers (EP0DTn to EP5DTn) are access registers used for read or write access into the transmission/receive buffer related to endpoint 0 to endpoint 5.

■ EP0 to EP5 Data Register (EP0DTn to EP5DTn)

Figure 63.3-12 EP0 to EP5 Data Register (EP0DTn to EP5DTn) The following describes the function of each bit in the EP0 to EP5 data registers (EP0DTn to EP5DTn)

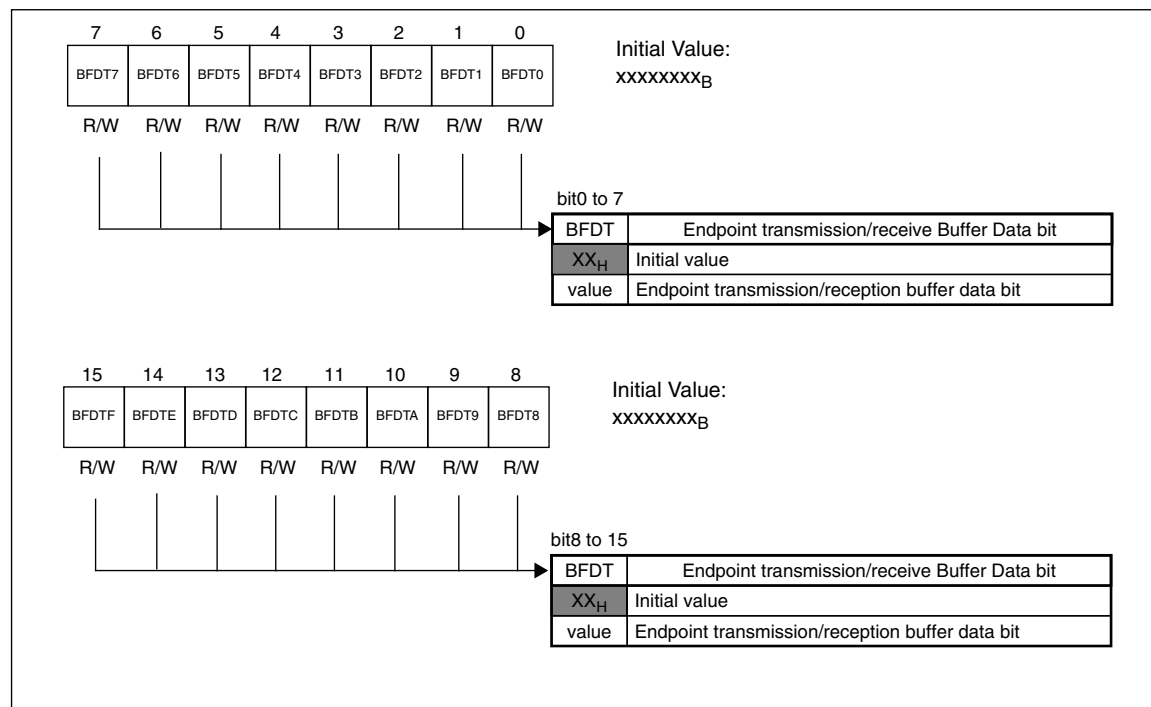


Table 63.3-12 Functional Description of each bit of the EP0 to EP5 Data Register (EP0DTn to EP5DTn).

Bit names		Function
Bit15-0	BFD7	<ul style="list-style-type: none"> It is a data read/write register for the transmission/receive buffer for each endpoint. Access to the BFD7 register via DMA transfer is supported by word access only. If you transfer odd number of data bytes through DMA transfer, you have to set a byte transfer for the last data transfer. If you perform word transfer (for odd number of data bytes) via CPU access, the last transfer must be a byte transfer, the same as in case of DMA transfer. CPU access to the EP0DTn to EP5DTn registers are possible both byte and word. If byte access is needed for any of the registers, access lower byte (bit 7 to bit 0) first and then access upper byte (bit 15 to bit 8). Subsequently access the lower byte and upper byte alternately. Bit access to the EP0DTn to EP5DTn registers is prohibited

63.4. Description of the USB function operation

This chapter describes the basics of the USB function.

■ USB Function operation

The USB function performs a both way packet transfer with a host controller that supports USB protocol. A host and its devices are connected and configured by the enumeration process. Then, communication based on various types of transfers using device drivers can be performed.

This section describes the operation of the USB communication between a HOST and the USB device by using enumeration as an example. It illustrates the operation of registers and USB packet transfers to give an overview of the USB communication.

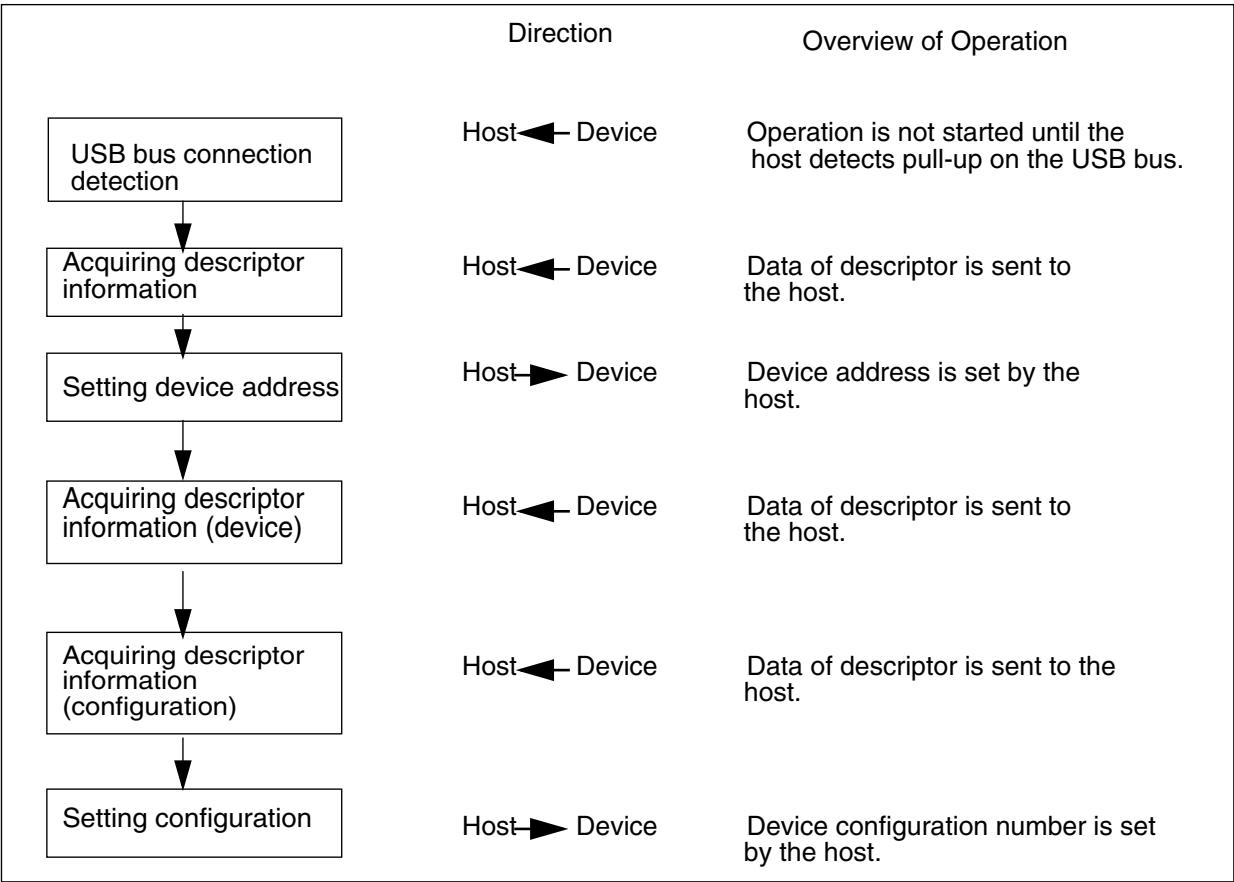
● Enumeration process

The enumeration process is the initial communication process between a host and the USB device when the device is attached to the bus before the USB operates. The host examines which device is connected to the USB bus and what parameters does it require, the number of EndPoints it has etc., by using the USB Control transfers (one of the types of USB transfers). USB Control Transfers use the EP0 (endpoint 0) out of the available six endpoints (as defined in the USB specification).

Before EP1 to EP5 are used, the followings must be received on the USB bus (i.e basic enumeration process should be completed).

- 1. USB bus reset (UDP is "0" and UDM is "0" for at least 40 USB clock cycles).
- 2. Address set by SET_ADDRESS.
- 3. Configuration set by SET_CONFIG.

Figure 63.4-1 Example of Connecting for USB Cable Terminal



● Detecting a connection

The HOST monitors the two signals (D+ and D-) on the USB bus and detects a device connection if a signal goes to the "H" level.

For the detailed procedure for the case of a self-powered device, see "[63.4.1 Detecting Connection and Disconnection](#)".

● Example of Register Initialization and Operation Startup Procedure

The following example describes how to initialize the registers and start operation.

1. Set EP0 in the EP0Cn register (packet size, etc.).
2. Set the EPEN, DIR, TYPE, etc. settings for each endpoint (see "[63.3.3 EP1 to EP5 Control Register \(EP1Cn to EP5Cn\)](#)").
3. Clear the RST bit in the UDCCn register.
4. Clear BFINI in the EP0ISn, EP0OSn, and EP1Sn to EP5Sn registers.
5. Clear the HCONX bit in the UDCCn register.

● USB bus reset

A bus reset is issued by the HOST to initialize the USB device. The USB device must then perform the following steps: (The first bus reset after USB has been connected does not need any processing.)

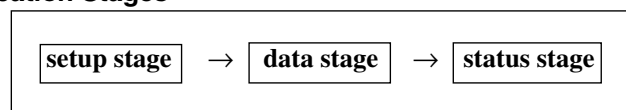
1. The USB function is initialized by setting the RST bit of the UDCCn register.
2. Set again the transmission/reception buffers in use and related control registers
3. Set the firmware control to the pre-enumeration state.

● Getting descriptor

The USB device sends data to the host when it receives a request from the HOST.

In more detail, communications are performed in the following three stages:

Figure 63.4-2 Communication Stages



The setup stage ensures that the device receives normal packets (specific request e.g. GET_DESCRIPTOR) from the HOST and identifies the command by decoding it and prepares information of a descriptor in the transmit buffer that is to be sent back to the host in Status stage. The data stage simply confirms that normal data is sent or received from the HOST (depending on whether it is a SET request or GET request). The status stage reports the status of the overall request and performs the end processing when the HOST sends (in response to the DESCRIPTOR sent in the data stage) a zero length data packet.

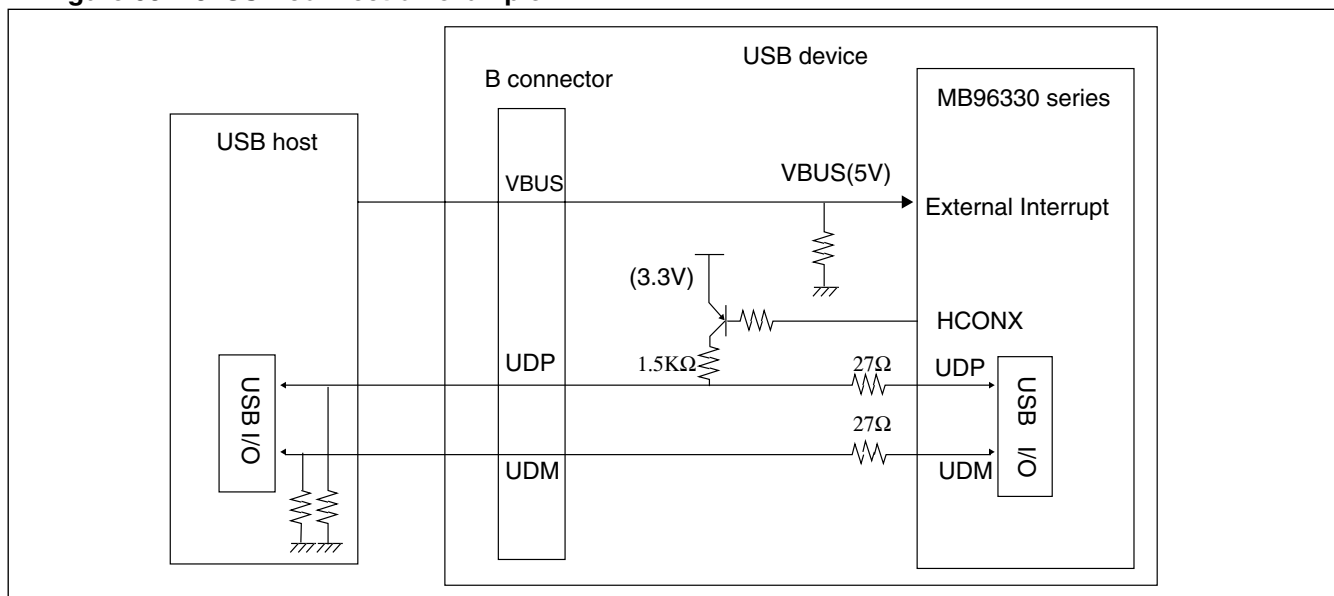
63.4.1 Detecting Connection and Disconnection

This section describes how to detect connection and disconnection from the USB host.

■ USB connection example

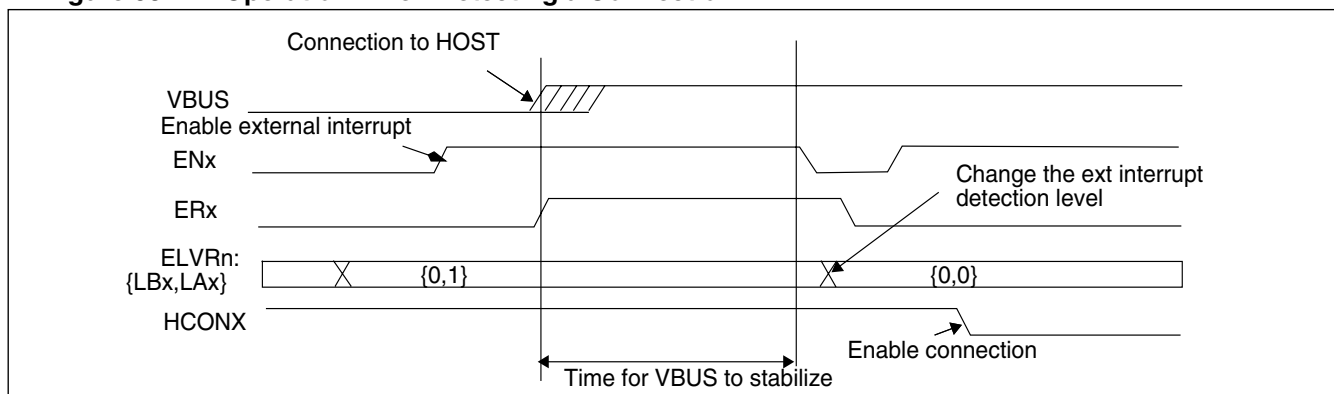
Connection and disconnection from the USB host can be detected by connecting an external interrupt pin to the VBUS pin on the USB connector and connecting a pull-down resistor. Figure 63.4-3 shows an connection example for the UDP, UDM, and VBUS pins on the USB connector.

Figure 63.4-3 USB connection example



● Detecting connection

Figure 63.4-4 Operation When Detecting a Connection



The device uses the following sequence to detect a connection with the HOST.

1. Set the external interrupt connected to the VBUS to detect "H" level input and then enable the interrupt.
2. Detection of an "H" level on the external interrupt pin indicates a connection to a USB host. When this occurs, please wait for the VBUS stabilization time before proceeding further.
3. Disable the external interrupt temporarily. Set the external interrupt to detect "L" level inputs to the external interrupt pin (for detecting disconnection from the host) and clear the external interrupt flag and enabled it again.
4. Initialization (complete initialization including the USB Function Register) can then be performed. See "I Example of Register Initialization and Operation Startup Procedure" in section 63.4. Description of the

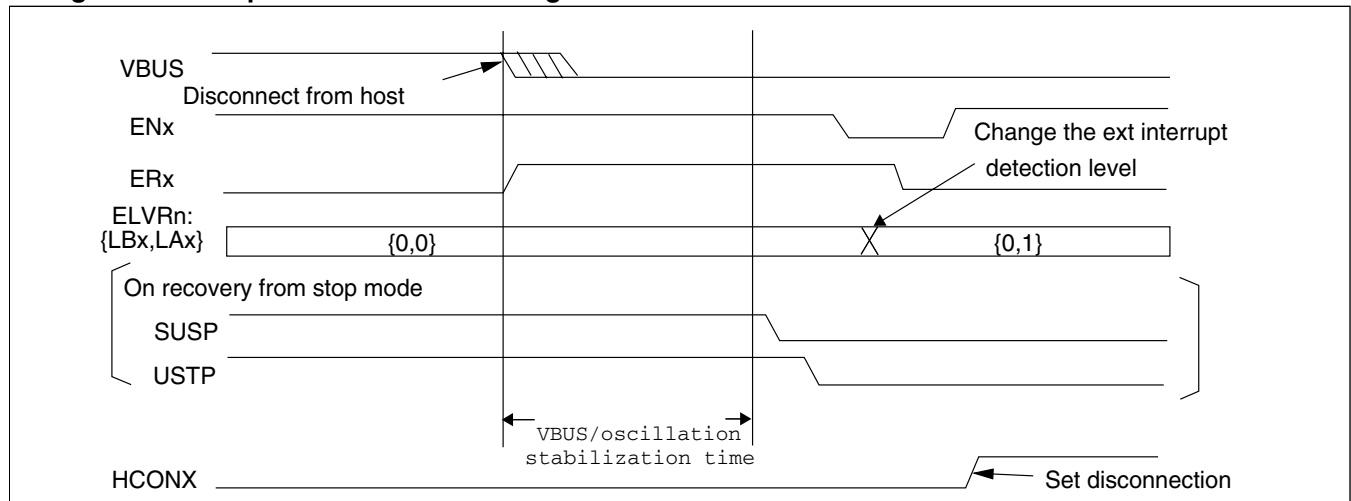
USB function operation.

5. Clear UDCCn:HCONX bit by writing "0" to it to enable the D+ pull-up resistor. (Clear the HCONX bit even if it is not used for controlling the pull-up resistor).

Note: You need not wait for the VBUS stabilization time in your program if an external noise filter is used on the external interrupt pin.

● Detecting disconnection

Figure 63.4-5 Operation When Detecting Disconnection



The USB device uses the following sequence to detect a disconnection from the HOST.

1. Detection of an "L" level on the external interrupt pin connected to VBUS indicates the disconnection from the USB host.
2. On recovery from STOP mode:
After waiting for the oscillation stabilization time, clear UDCCn:SUSP followed by UDCCn:USTP.
When not recovering from stop mode:
Wait for the VBUS stabilization time.
3. Disable the external interrupt temporarily. Change the external interrupt setting to detect "H" level inputs to the external interrupt pin, clear the external interrupt flag, and then re-enable the external interrupt.
4. Set the UDCCn:HCONX bit to "1" to disconnect the D+ pull-up resistor. (Set the HCONX bit to "1" even if it is not used for control of the pull-up resistor).

Note: You need not wait for the VBUS stabilization time in your program if an external noise filter is used on the external interrupt pin.

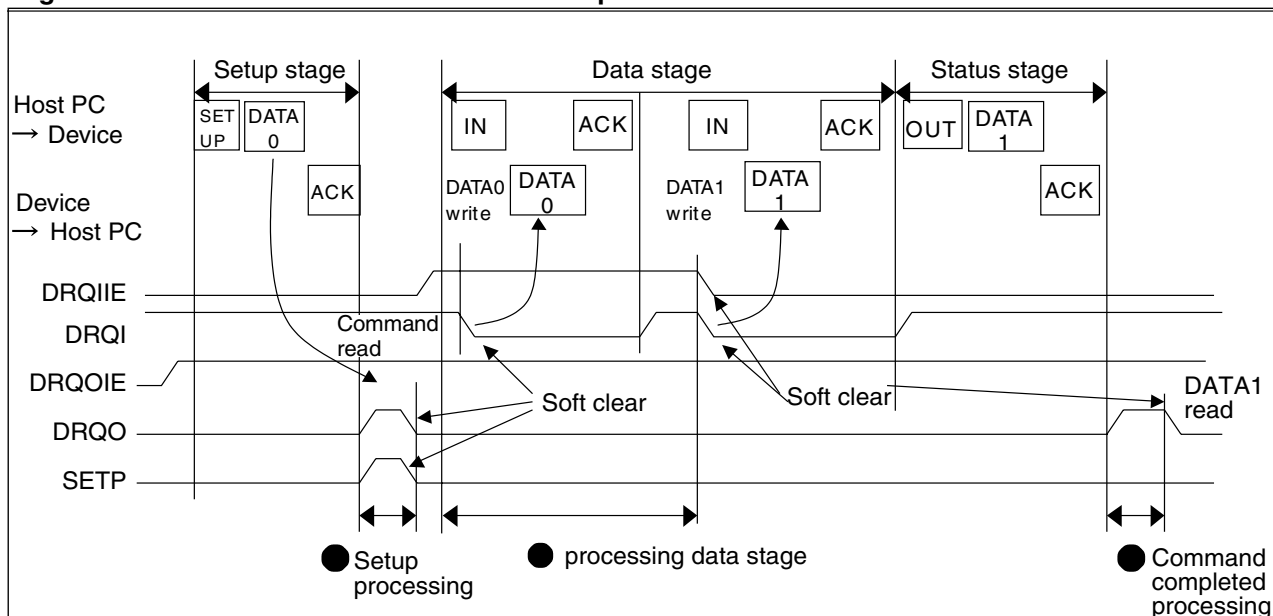
63.4.2 Each Register Operation with Command Response

This section describes the basic operation of USB registers and processing of the USB packets. The processing of firmware tasks triggered via CPU interrupts are also described.

■ Read Command USB function operation

For GetDescriptor, SynchFrame, and the class vendor command

Figure 63.4-6 Read Command USB function operation



● Set up operation

When the setup packet is received by the USB device, DRQO and SETP are set to "1". When DRQO is set, CPU interrupt is raised and the SETP flag is confirmed in the interrupt service routine. The USB device then reads as many commands as necessary from the receive buffer if the SETP is set (which does not mean all eight bytes need to be read. The number of bytes to be read depends on the SIZE value in the EP0OSn register), decodes the commands, performs setting tasks, clears the SETP flag and DRQO interrupt cause (EP0OSn:DRQO) by writing "0" to them and returns from the interrupt service routine.

● Data stage operation

If the result of decoding the command says that the data stage is an IN transfer, enable the DRQIE bit (no need to set DRQI bit because its initial value is "1"). The CPU interrupt service routine for DRQI interrupt transfers data to be sent to the host to the transmit buffer. Once the transfer is complete, clear the DRQI interrupt cause (EP0ISn:DRQI) before returning from the interrupt.

The DRQI is set again when the data packet transfer to the host is complete. This generates a CPU interrupt and the corresponding interrupt service routine for DRQI is entered again. The data is transferred to the transmit buffer to prepare for the next data packet to be sent to the host. Once the transfer is complete, clear the DRQI interrupt cause (EP0ISn:DRQI) before returning from the interrupt service routine.

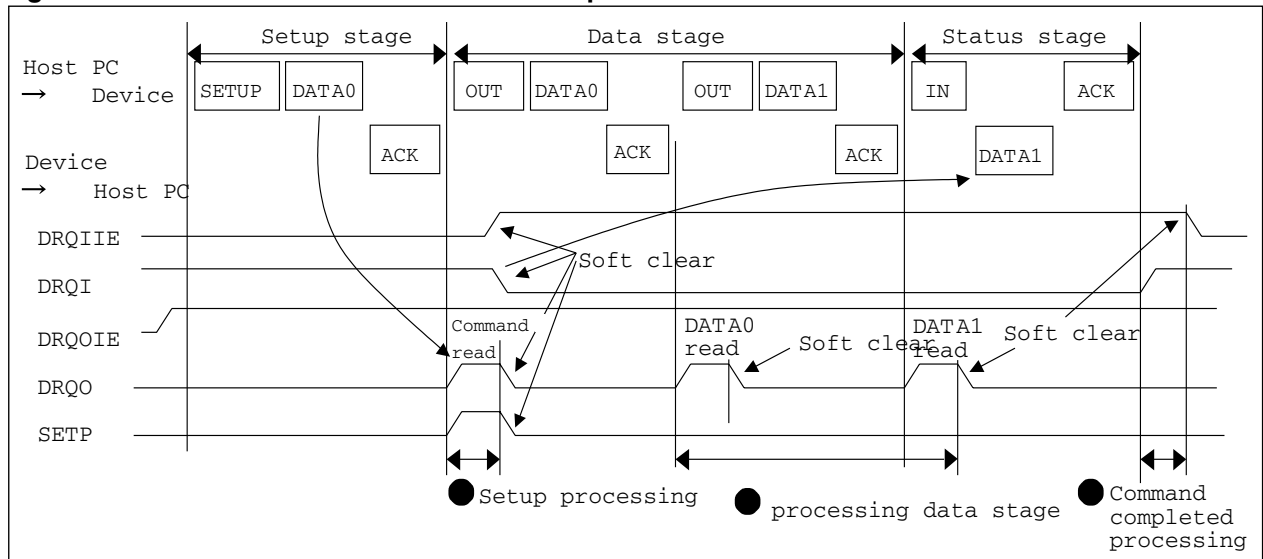
● Status stage operation

The DRQI is set when the data stage is complete. Since the DRQI interrupt is not enabled (DRQIE = "0") the CPU does not execute the interrupt service routine for DRQI. When a zero length OUT packet is received from the host the DRQO is set to "1". The CPU processes the DRQO interrupt and the number of received bytes is checked in the SIZE bits of the EP0OSn. If EP0Sn:PKS is equal to zero (indicating that a zero byte data packet is received) clear the DRQO interrupt cause to prepare for the next setup stage.

■ Write Command USB function operation

For GetDescriptor and the class vendor command

Figure 63.4-7 Write Command USB function operation



● Set up operation

When the setup packet is received, DRQO and SETP is set. When DRQO is set the CPU interrupt corresponding to DRQ0 is raised and the SETP flag is conformed in the interrupt service routine. The USB device then reads as many commands bytes as necessary from the receive buffer if the SETP is set (which does not mean all eight bytes need to be read. The number of bytes to be read depends on the SIZE value in the EP0OSn register), decodes the commands, performs setting tasks, and clear the DRQI bit (because initial value of DRQI is "1") without writing data to the transmission buffer to prepare for a 0-byte response in the status stage. Then set DRQIIE to enable the interrupt cause due to DRQI to check the normal completion of Status stage. The routine also clears the SETP flag and DRQO by writing "0" to it and then returns from the interrupt routine.

● Data stage operation

The DRQO is set when the data stage toward OUT is complete. The interrupt is processed when the DRQO is set. Then check the SIZE bits of the EP0OSn register and transfer the data from the receive buffer by DMA or read the data from the receive buffer directly by CPU. Then clear the DRQO bit before returning from the interrupt.

● Status stage operation

When the Data stage is completed the host sends an IN token request. A zero length packet is sent by the device in response to this request. When the status stage is completed, a CPU interrupt USB_EP0IN is triggered due to DRQI and the program enters in an interrupt service routine and confirms if the status stage has been successfully completed. Then clear the EP0ISn:DRQIIE before returning to the interrupted point.

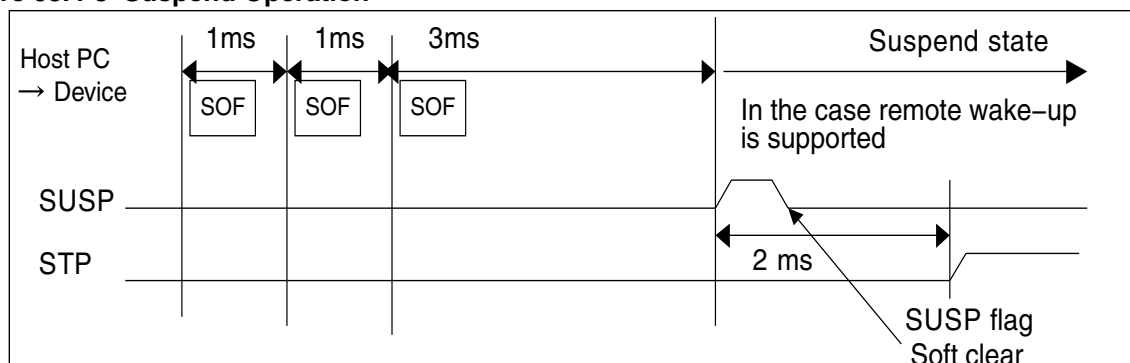
63.4.3 Suspend Mode Function

A USB device must have bus power supply configuration with a power consumption less than 500 μ A in SUSPEND mode. This section describes the USB device procedure to enter SUSPEND mode and then STOP mode.

■ USB function SUSPEND mode

When the USB device core detects a SUSPEND mode request, the SUSP flag of the UDCCSn register is set. The figure below shows an example of the suspend operation:

Figure 63.4-8 Suspend Operation



● SUSPEND mode operation

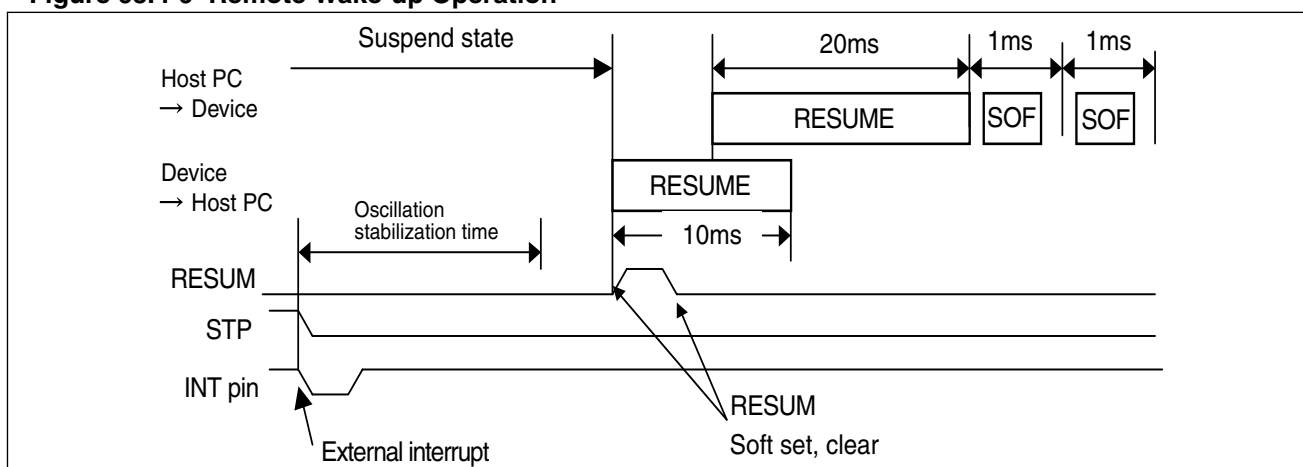
The USB function detects a SUSPEND mode when there is no activity for at least 3 ms on the bus. If so, the SUSP flag in the UDCCSn register is set which causes an interrupt. If a USB device supports remote wake-up, wait for another 2 ms (which blocks remote wake-up during this time period), and then set the usb device to STOP mode.

63.4.4 Wake-up Function

To change a USB device from SUSPEND mode to WAKE-UP mode, the USB protocol provides two different options: Remote wake-up from device and wake-up from HOST.

■ Remote wake-up by device

Figure 63.4-9 Remote Wake-up Operation

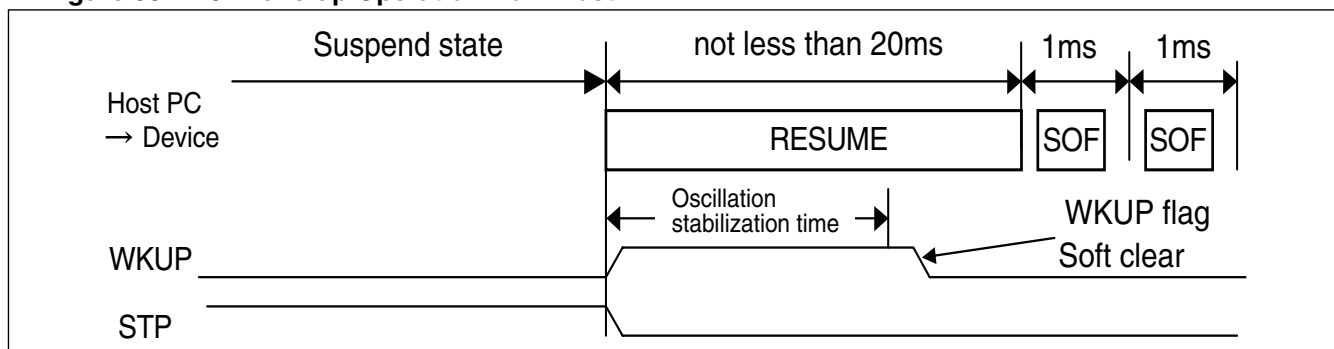


A USB device must performed the following steps:

1. Recover the USB device from STOP mode through an external interrupt.
2. Set the UDCCn:RESUM bit.
3. Clear the UDCCn:RESUM bit.

Wake-up from Host

Figure 63.4-10 Wake-up Operation from Host



For the devices, the following steps are required:

1. Oscillation stabilizing time setting must not exceed 10 ms.
2. WKUP interrupt triggers the entry into an interrupt routine that clears the WKUP flag of UDCSn (interrupt cause) and then returns to main program.

63.4.5 DMA Transfer Function

Data transfers between USB function buffers and internal RAM is possible with two different DMA transfer modes: Packet Transfer mode and Data Number Automatic Transfer mode. In Packet Transfer mode the DMA transfer data size is to be set for each packet one by one, on the other hands, in Data Number Automatic Transfer mode the DMA transfer data size for all packets is to be set once.

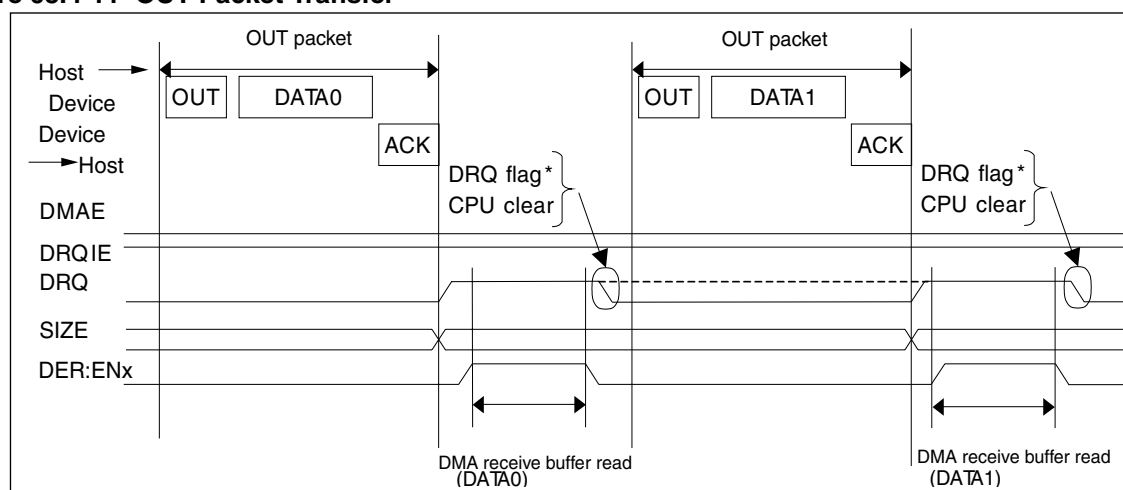
Packet Transfer Mode

The packet transfer mode performs transfer by setting the number of data per packet to be transferred by the DMA and clearing the interrupt cause when the transfer is complete. The transfer mode can access any endpoint buffer.

Buffer access timings in OUT and IN directions are described below.

OUT direction (HOST → device) transfer

Figure 63.4-11 OUT Packet Transfer



In OUT- direction transfer, the USB device performs the following steps:

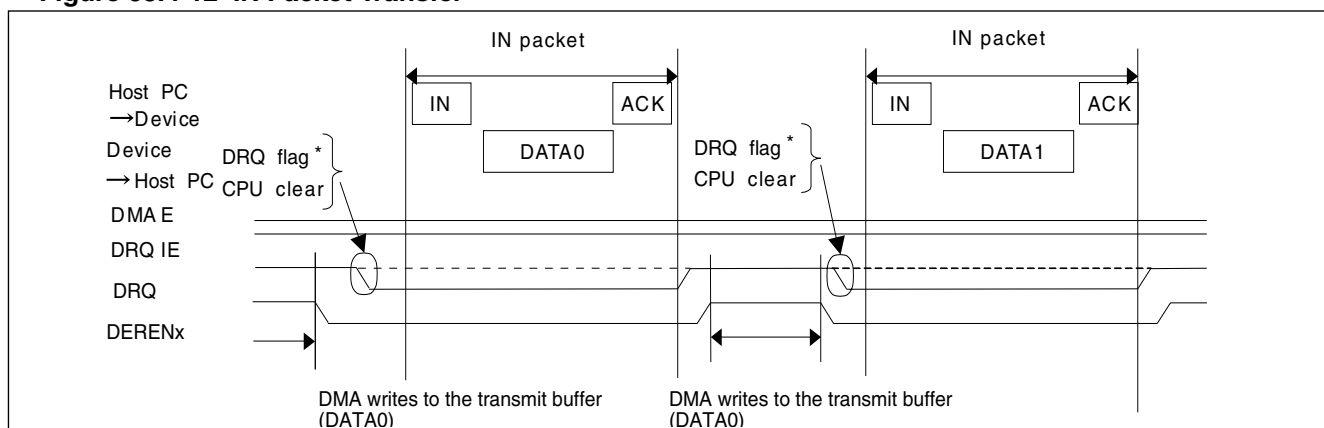
1. When the DRQ flag is set, the interrupt routine is called and the number of received data bytes must be checked (in EPxSn:PKS)
2. Set the number of data to be transferred in the data counter register DCT of DMA and the DMA is enabled by setting the DER register.

- Once the transfer is complete, clear the corresponding DRQ flag in the EPxSn registers and the corresponding flag in the DSR register of the DMA.

Note: Each of the endpoints EP1 to EP5 consists of double buffers. DRQ can be cleared only when one buffer that is currently not being accessed is empty and data has been read from the other buffer. It cannot be cleared (even if "0" is written to it) when the buffer which is not being accessed has data left to be read (Dotted line status). In this case it continuously enters DRQ interrupt process.

- IN direction (device → host) transfer.

Figure 63.4-12 IN Packet Transfer



In IN- direction transfer, a USB device performs the following steps:

- Once the DRQ flag is set, the device enters the interrupt routine in which the number of data to be transferred in an IN packet is set in the data counter register DCT of the DMA. It enables the DMA by setting the DER register and data transfer from RAM to EndPoint buffer is triggered.
- Once the DMA transfer is complete, clear the corresponding DRQ flag in the EP1Sn to EP5Sn registers and the corresponding interrupt flag in the DSR register of the DMA . The device then returns from the interrupt process.

Note: Each of endpoints EP1 to EP5 consists of double buffers. DRQ can be cleared only when one buffer that is currently not being accessed has already data written into it and data has been written to the other buffer. It cannot be cleared (even if "0" is written to it) when the buffer which is not being accessed is empty (dotted line status). In this case it continuously enters DRQ interrupt process.

■ Data Number automatic Transfer Mode

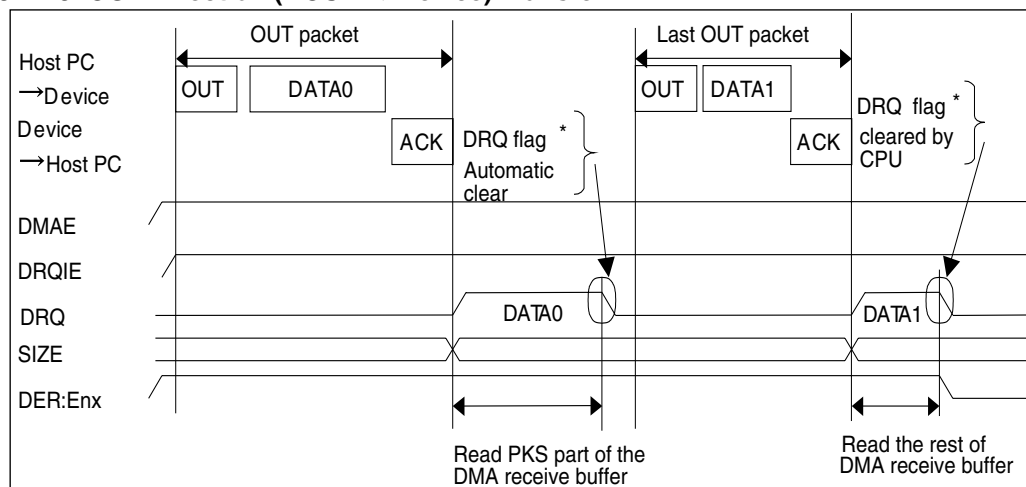
In this mode, the total number of data to be transferred is set in DMA and the transfer enable bit is set in advance. When EPxCn:DMAE bit is enabled and the DRQ is set after data from the Host is received successfully (data is ready to be read from the receive buffer), DRQ is automatically cleared after the data equal to the number of bytes indicated in the EPxSn:PKS is transferred. (Whether the DRQ flag is actually cleared depends on the fact that both buffers in the double buffer are empty or full). Similar process is repeated until the exact number of data to be transferred as defined in the DMA have been effectively transferred except for the last packet transfer. For the last packet transfer under the Data Number Automatic Transfer mode, the DRQ interrupt cause is not cleared automatically but is cleared by CPU in the interrupt service routine.

If the device performs the next transfer, it sets DMA again and enables DMA when a CPU interrupt is raised after the last data has been transferred, and returns from the CPU interrupt. Since the data number automatic transfer mode is used for DMAE=1, only buffer access to endpoint 1 to 5 is enabled.

Buffer access timings in OUT/IN directions are described below:

● OUT direction (HOST → device) transfer

Figure 63.4-13 OUT Direction (HOST → Device) Transfer



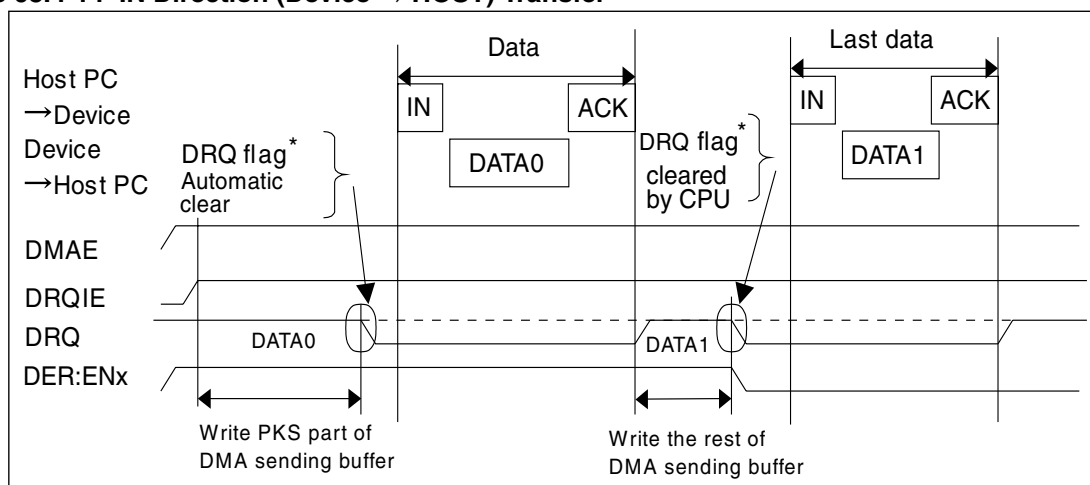
In OUT transfer directions, a USB device performs the following steps:

1. Set the total number of data to be transferred in the data counter register DCT in DMA, and enable the DMA by setting the DER register.
2. Set EPxCn:DMAE and EPxSn:DRQIE.
3. Once the transfer is complete, a DMA interrupt routine is called and clear the flag (DRQ and DTE_x) and to set the DMA for next transfer, and then return from the interrupt routine.

Note: Each of the endpoints EP1 to EP5 consists of double buffers. DRQ can be cleared only when one buffer that is currently not being accessed is empty and data has been read from the other buffer (automatic clear). It cannot be cleared (even if "0" is written to it) when the buffer which is not being accessed has data left to be read (Dotted line status). In this case it continuously enters DRQ interrupt process.

● IN direction (HOST → device) transfer

Figure 63.4-14 IN Direction (Device → HOST) Transfer



In IN transfer direction, a USB device performs the following steps:

1. Set the total number of data to be transferred in the data counter register DCT in DMA, and enable DMA by setting the DER register.
2. Set DMAE and DRQIE.
3. Once the transfer is complete, a DMA interrupt routine is called, then set DMA again and clear the flags,

then return from the interrupt routine.

Note: Each of endpoints EP1 to EP5 consists of double buffers. DRQ can be cleared only when one buffer that is currently not being accessed has already data written into it and data has been written to the other buffer. It cannot be cleared (even if "0" is written to it) when the buffer which is not being accessed is empty (dotted line status). In this case it continuously enters DRQ interrupt process.

63.4.6 NULL Transfer Function

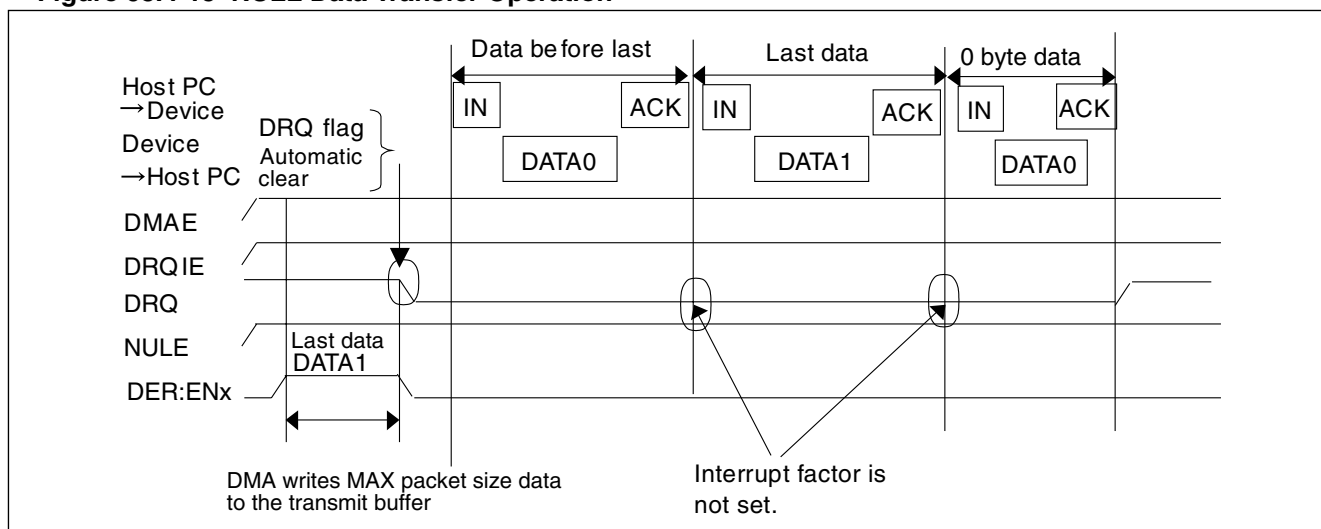
In case of Data Number Automatic IN type transfer if the total number of data bytes as set in the DCT register of DMA is already sent, its possible to automatically send a 0-byte data packet in response to the next IN transfer request from the host by enabling the NULL transfer mode.

■ NULL Transfer Mode

In this mode the 0 byte data transfer is set automatically when the last IN transfer request from the HOST is detected in the case the Data number automatic transfer mode is set for IN transfers (DMAE=1) and the maximum size data set in PKS bits is written in to the transfer buffer. If the DCT register of DMA is decremented to 0 at the last data writing, a 0 byte data is transferred when the next IN transfer request from the host is detected. The DRQ interrupt flag is not set until 0-byte data is received by the host i.e. after receiving the ACK for the 0-byte transfer from the host. Buffer access timings are shown in the figure below.

- Only IN direction (device → host) transfer

Figure 63.4-15 NULL Data Transfer Operation



For the device, DMAE, DRQIE, and NULE bits must be set.

MB91460 Series

Chapter 64 ShutDown Control

MB91460E series contains a new low power control to minimize leakage current in STOP and RTC mode.

64.1. Overview

In Shutdown mode, the power supply of more than 80% of the internal logic and the main memories is switched off to minimize leakage.

This mode is a type of STOP state.

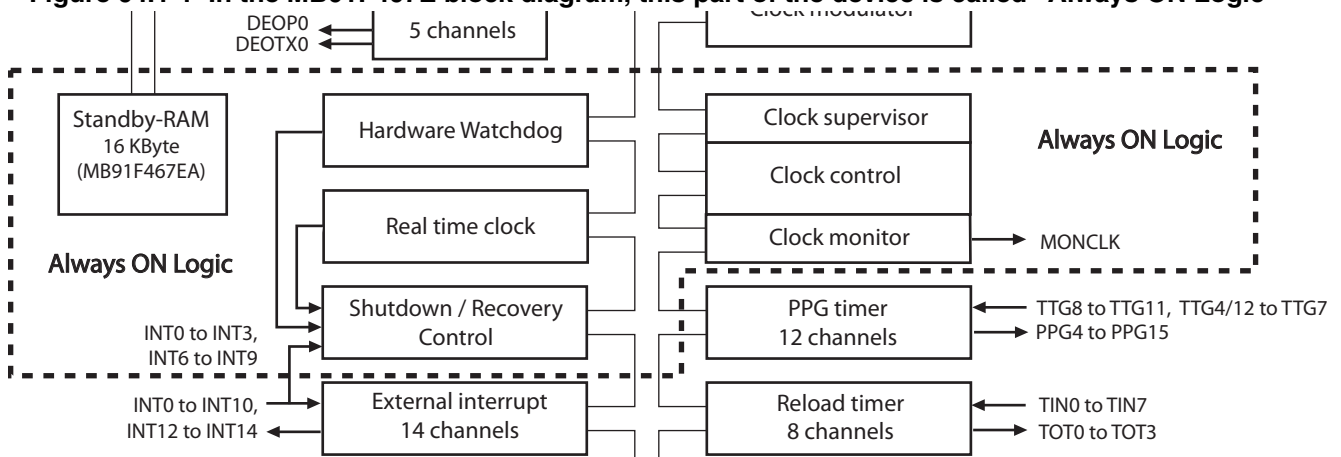
The device can enter this mode if it goes to STOP state when Shutdown is enabled.

During this mode, the oscillators can stop oscillating and the power is not supplied except for some logic.

The power continues to be supplied to the following circuits even in shutdown state:

- Standby RAM 16 KByte for data (address FFFAC000H to FFFAFFFH)
- Shutdown / recovery control circuit
- Clock control logic
- Real Time Clock
- 4 MHz oscillator + 32 kHz oscillator + RC oscillator
- Hardware Watchdog + Clock Supervisor

Figure 64.1-1 In the MB91F467E block diagram, this part of the device is called “Always ON Logic”



The device will recover from Shutdown mode after the following events:

- Reset assertion by the INITX pin
- External interrupt (8 sources)
- Real Time Clock interrupt
- Hardware watchdog reset
- Main Clock Supervisor reset

64.2. Standby RAM

MB91F467E contains a 16 KByte low-leakage RAM used as Standby RAM. The power supply of this RAM is not switched off in Shutdown state.

The Standby RAM is located at addresses FFFAC000H to FFFAFFFH.

To access it, to the RAM must be enabled by setting RAMEN bit in SHDE register. RAMEN is initialized by Software Reset (RST). If the RAM is to be accessed, make sure that no external bus Chip Select area

overlaps the Standby RAM addresses.

For the Standby RAM, low-leakage macros have been implemented. Read and write acces are performed with 1 wait cycle.

64.3. Shutdown Registers

64.3.1 Notes About the Reset Signals

The following register description mentions different reset signals, which are explained shortly here. For more information, please refer to the MB91460 series hardware manual, “Chapter 9 Reset”.

- **Settings Initialization Reset (INIT):**
initializes all the device's control and clock settings. INIT can be triggered
 - by low level on external INITX pin
 - by low level on external HSTX pin (no hardware standby pin available in MB91460E series)
 - by Hardware Watchdog Timer
 - by Clock Supervisor
 - by Software Watchdog Timer
 - by Low Voltage Detection
- **Operation Initialization Reset (RST, “Software Reset”):**
initializes CPU and peripherals and restarts the software. RST can be triggered
 - by low level on external RSTX pin (not available in MB91460E series)
 - by INIT (INIT always causes RST)
 - by software (STCR.SRST=0)
- **Shutdown Recovery:** The Shutdown state is released when a valid recovery factor is found. Shutdown recovery causes a Settings Initialization Reset (INIT) with some exceptions. For details, please refer to [64.4.2 Recovery from shutdown mode \(Page No.1642\)](#).

64.3.2 SHDE: Shutdown control register

This register enables/disables the shutdown state as well as the Standby RAM.

■ **SHDE : Address 0004D4_H Access: Byte**

	7	6	5	4	3	2	1	0	
	SDENB	-	-	-	-	-	-	RAMEN	
0	X	X	X	X	X	X	X	0	Initial value ¹
retained	X	X	X	X	X	X	X	0	Initial value ²
R/W	-	-	-	-	-	-	-	R/W	Attribute

1. Initial value after external pin INITX=0 or Shutdown Recovery
2. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit 7: **SDENB** Shutdown enable

SDENB	Function
1	Enable shutdown state: On transition to STOP mode, the device enters Shutdown state.
0	Disable shutdown state: On transition to STOP mode, the device enters the normal STOP mode.

MB91460 Series

- bit 6-1: Reserved bits
The read value is undefined. Always write 0 to these bits.
- bit 0: **RAMEN** Standby RAM enable

RAMEN	Function
1	Enable the Standby RAM ¹ : Read and write access to the Standby RAM is possible
0	Disable the Standby RAM: Read and write access to the Standby RAM is disabled.

1. The Standby RAM is located inside the address space of External Bus. If the Standby RAM is enabled, make sure that no chip select area of the External Bus overlaps the standby RAM area.

Note: **RAMEN** is cleared by INIT and by Software Reset because the chip select control registers (CSER, ACR0-7, ASR0-7, AWR0-7) are initialized by the same conditions. After both kinds of reset, chip select CS0 is enabled to cover all addresses of external bus area, which would overlap the Standby RAM address space.

64.3.3 EXTE: Shutdown recovery external interrupt enable register

This register enables external interrupts as the source for recovering from the shutdown state.

■ **EXTE : Address 0004D6_H Access: Byte**

7	6	5	4	3	2	1	0	
RX1	RX0	INT7	INT6	INT3	INT2	INT1	INT0	
0	0	0	0	0	0	0	0	Initial value ¹
retained	retained	retained	retained	retained	retained	retained	retained	Initial value ²
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

1. Initial value after external pin INITX=0 or Shutdown Recovery
2. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

Eight external interrupts that can be set as recovery sources are allocated to each bit, as shown in the [Table 64.3-1](#):

Table 64.3-1 External interrupts which can be set as recovery sources

bit	Pin No ¹	Pin Name
7	93	P32_2/RX1/INT9
6	91	P23_0/RX0/INT8
5	90	P24_7/SCL3/INT7
4	89	P24_6/SDA3/INT6
3	86	P24_3/INT3
2	85	P24_2/INT2
1	84	P24_1/INT1
0	83	P24_0/INT0

1. The pin numbers of MB91F467E are listed here.

- bit 7-0 Interrupt enable bits

Value	Function
1	Enable recovery interrupt
0	Disable recovery interrupt

These bits can be read and written.

External pin INITX=0 or Shutdown recovery clear these bits.

64.3.4 SHDINT: Shutdown recovery internal interrupt control and status register

The SHDINT register contains control bits and flags for enabling and indicating internal interrupts for recovery from shutdown mode.

■ SHDINT : Address 0004DB_H Access: Byte

7	6	5	4	3	2	1	0	
-	-	-	-	HWWDF	HWWDE	RTCF	RTCE	
X	X	X	X	0	0	0	0	Initial value ¹
X	X	X	X	retain	0	retain	0	Initial value ²
X	X	X	X	retain	0	retain	retain	Initial value ³
-	-	-	-	R(RM1)/ W0	R	R(RM1)/ W0	R/W	Attribute

1. Initial value after external pin INITX=0
2. Initial value after Shutdown Recovery
3. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

- bit 7-4 Reserved bits

The read value is undefined. Always write 0 to these bits.

- bit 3 **HWWDF**: Hardware Watchdog recovery flag

HWWDF	Function
1	Recovery factor from Hardware Watchdog found
0	No recovery factor from Hardware Watchdog found

This bit is set in Shutdown mode, if HWWDE is set and if an INITX signal from Hardware Watchdog is detected.

Writing "1" to this bit does not affect the operation.

Writing "0" clears the bit, external pin INITX=0 clears the bit.

"1" is read by a read-modify-write instruction.

MB91460 Series

- bit 2 **HWDE**: Hardware Watchdog recovery enable (mirror of HWWDE.STP_RUN ¹⁾)

HWWDE	Function
1	Recovery reset from Hardware Watchdog is enabled, RC clock is enabled in STOP/Shutdown mode by hardware
0	Recovery reset from Hardware Watchdog is disabled, RC clock depends on CSVCR.RCE setting in STOP/Shutdown mode

This bit is a read-only mirror of HWWDE.STP_RUN, which can be set only once after reset and cannot be cleared by CPU access.

This bit is cleared by Software Reset (RST). Note that external pin INITX=0 or Shutdown recovery are always followed by a Software Reset RST.

- bit 1 **RTCF**: Real Time Clock recovery flag

RTCF	Function
1	Recovery factor from Real Time Clock found
0	No recovery factor from Real Time Clock found

This bit is set in Shutdown mode, if RTCE is set and an interrupt signal from Real Time Clock is detected.

Writing "1" to this bit does not affect the operation.

Writing "0" clears the bit, external pin INITX=0 clears the bit.

"1" is read by a read-modify-write instruction.

- bit 0 **RTCE**: Real Time Clock recovery enable

RTCE	Function
1	Recovery reset from Real Time Clock is enabled
0	Recovery reset from Real Time Clock is disabled

This bit can be read and written.

External pin INITX=0 or Shutdown recovery clear this bit.

1. STP_RUN is bit HWWDE[4]

64.3.5 EXTF: Shutdown recovery external interrupt source flags

This register indicates the recovery source for when a shutdown recovery external interrupt is used to recover.

■ **EXTF : Address 0004D7_H Access: Byte**

7	6	5	4	3	2	1	0	
RX1	RX0	INT7	INT6	INT3	INT2	INT1	INT0	
0	0	0	0	0	0	0	0	Initial value ¹
retained	retained	retained	retained	retained	retained	retained	retained	Initial value ²
retained	retained	retained	retained	retained	retained	retained	retained	Initial value ³
R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	R(RM1)/ W0	Attribute

1. Initial value after external pin INITX=0
2. Initial value after Shutdown Recovery
3. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

The bit configuration is the same as for the EXTE register.

- bit 7-0 Interrupt factor flag bits

The bit corresponding to any input signal found to be valid as a recovery factor is set to "1."

Value	Function
1	Recovery factor found
0	No recovery factor found

These bits are set in Shutdown mode, when the attached external interrupt channel is enabled by EXTE=1 and a recovery factor (level / edge) from the external interrupt channel is detected.

Writing "1" to these bits does not affect the operation.

Writing "0" clears the bits, external pin INITX=0 clears the bits.

"1" is read by a read-modify-write instruction.

MB91460 Series**64.3.6 EXTLV1/2: Shutdown recovery external interrupt level selection register**

This register sets the pin level for recovering from the shutdown state using an external interrupt.

■ **EXTLV1 : Address 0004D8_H Access: Halfword, Byte**

15	14	13	12	11	10	9	8	
LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	
0	0	0	0	0	0	0	0	Initial value ¹
retained	retained	retained	retained	retained	retained	retained	retained	Initial value ²
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

1. Initial value after external pin INITX=0 or Shutdown Recovery

2. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

■ **EXTLV2 : Address 0004D9_H Access: Halfword, Byte**

7	6	5	4	3	2	1	0	
LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	
0	0	0	0	0	0	0	0	Initial value ¹
retained	retained	retained	retained	retained	retained	retained	retained	Initial value ²
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

1. Initial value after external pin INITX=0 or Shutdown Recovery

2. Initial value after Software Reset (RST)

(For attributes, refer to “[Meaning of Bit Attribute Symbols \(Page No.15\)](#)”.)

Table 64.3-2 shows the allocation of the source level control pins of eight external interrupts that can be set as recovery sources:

Table 64.3-2 Source level control bits of the external shutdown recovery interrupt pins

bit	Pin No ¹	Pin Name
15,14	93	P23_2/RX1/INT9
13,12	91	P23_0/RX0/INT8
11,10	90	P24_7/SCL3/INT7C
9,8	89	P24_6/SDA3/INT6D
7,6	86	P24_3/INT3
5,4	85	P24_2/INT2
3,2	84	P24_1/INT1
1,0	83	P24_0/INT0

1. The pin numbers of MB91F467E are listed here.

- bit 15-0: Interrupt level setting register

LBx	LAx	Interrupt Level
0	0	"L" level (initial value)
0	1	"H" level
1	0	Rising edge
1	1	Falling edge

Please refer to [External Interrupts: Level or Edge Setting \(Page No.1641\)](#).

64.4. Shutdown Operation

64.4.1 Transition to shutdown state

Shutdown state is a special kind of the STOP state. During Shutdown, the settings in the STCR register for Oscillation Disable (STCR.OSCD1, STCR.OSCD2), Hi-Z mode (STCR.HIZ) and Oscillation Stabilization time (STCR.OS[1:0]) are valid the same kind as in normal STOP state. At recovery from Shutdown, STCR.OS[1:0] are not cleared to maintain the oscillator stabilisation time, while STCR.OSCD1, STCR.OSCD2 and STCR.HIZ are initialized by the recovery.

For transition into Shutdown, do the following:

- Enable at least one recovery condition (otherwise, recovery is only possible by external INITX pin)
- Enable the Shutdown mode
- Switch the device to STOP mode
- The details are explained below.

■ Precautions

Before enabling Shutdown, consider the following:

- Data, which is needed after recovery from Shutdown, should be copied into the Standby RAM.
- The CPU should run on Main- or Sub-Oscillation, not on PLL. The PLL should be disabled.
- The Sub-Regulator can be set to 1.2V in STOP mode by setting REGSEL.SUBSEL = 0x00
- Specify the levels of external interrupt signals used for recovery in EXTLV1/2 registers
- Enable the channels of external interrupt signals for recovery in EXTE register

■ Deep Shutdown Settings for maximal power saving

The following settings generate Shutdown without any activity on the device:

- Disable all pin pull-up/pull-down settings which are not required, or set the STCR.HIZ¹ bit when going to STOP.
- Set external bus pins to port mode / input direction (otherwise some pins will output constant values, see [64.4.5 I/O Behaviour in Shutdown \(Page No.1645\)](#)).
- Don't set Hardware Watchdog Run in STOP mode (HWWDE.STP_RUN² =0, this is default setting)
- Disable the RC oscillator in STOP mode (CSVCR.RCE=0)
- Disable the Low Voltage Detection in STOP mode (LVDET.LVEPD=1, LVDET.LVIPD=1)
- Disable the Main and the Sub oscillators in STOP mode (STCR.OSCD1=1, STCR.OSCD2=1)
- Set the Shutdown Enable bit SHDE.SDENB=1 to enable shutdown mode

-
1. With STCR.HIZ=1, all pull-ups and pull-downs are disabled in STOP/Shutdown.
 2. STP_RUN is bit [4] of HWWDE register. It enables running the Hardware Watchdog in STOP mode. STP_RUN can only be set by software, but not cleared. STP_RUN is cleared by INIT.

MB91460 Series

- Go to STOP: set the STOP request `STCR.STOP=1` and read back `STCR` two times.

■ Shutdown with Real Time Clock running

The following settings generate Shutdown with the RTC running on Main-Oscillation, Sub-Oscillation or RC clock, and with recovery by RTC enabled:

- Set the RTC prescaler values depending on the clock speed (`WTBR` register)
- If recovery by the RTC is needed:
 - Enable at least one of the the RTC interrupts (half-second, second, minute, hour or day) in `WTCR` and/or `WTCE` register
 - Enable RTC recovery: set `SHDINT.RTCE=1`
- If RTC uses Main Oscillation:
 - Disable the RC oscillator in STOP mode (`CSVCR.RCE=0`)
 - Disable the Sub oscillator in STOP mode (`STCR.OSCD2=1`) and keep Main oscillator running (`OSCD1=0`)
 - The RTC is connected to Main oscillation by default.
- If RTC uses Sub Oscillation:
 - Disable the RC oscillator in STOP mode (`CSVCR.RCE=0`)
 - Disable the Main oscillator in STOP mode (`STCR.OSCD1=1`) and keep Sub oscillator running (`OSCD2=0`)
 - Connect the RTC to Sub oscillator: set `CSCFG.CSC[1:0]=01`
- If RTC uses RC clock:
 - Enable the RC oscillator in STOP mode (`CSVCR.RCE=1`, this is default setting)
 - Disable the Main and the Sub oscillators in STOP mode (`STCR.OSCD1=1`, `STCR.OSCD2=1`)
 - Connect the RTC to RC oscillator: set `CSCFG.CSC[1:0]=10`
- Set the Shutdown Enable bit `SHDE.SDENB=1` to enable shutdown mode
- Go to STOP: set the STOP request `STCR.STOP=1` and read back `STCR` two times.

■ Hardware Watchdog in Shutdown

The Hardware Watchdog can run in STOP mode, if the bit `HWWDE.STP_RUN`¹ is set.

- Outside STOP mode, the Hardware Watchdog timeout will send an INIT signal to the CPU via the Shutdown control.
- In STOP mode without Shutdown, the Hardware Watchdog timeout will send an INIT signal to the CPU via the (inactive) Shutdown control, which cancels the STOP mode immediately.
- In STOP mode with Shutdown enabled, the Hardware Watchdog timeout will set the `SHDINT.HWWDF` flag, causing a recovery from Shutdown.

The Hardware Watchdog can be enabled in Shutdown state like follows:

- Enable the Hardware Watchdog operation in STOP mode: set `HWWDE.STP_RUN = 1`
In parallel, this enables the RC oscillator by hardware, and the Hardware Watchdog recovery Enable bit `SHDINT.HWWDE` is set by hardware too.
- If RTC is needed, enable it like described in n“Shutdown with Real Time Clock running” above.
- Specify the levels of external interrupt signals used for recovery in `EXTLV1/2` registers
- Enable the channels of external interrupt signals for recovery in `EXTE` register
- Set the Shutdown Enable bit `SHDE.SDENB=1` to enable shutdown mode
- Clear/restart the Hardware Watchdog: write 0 to bit `HWWD.CL`
- Go to STOP: set the STOP request `STCR.STOP=1` and read back `STCR` two times.

If the timeout is reached, the Hardware Watchdog generates INIT, which cancels the Shutdown state and forces recovery. The CPU will run on Main Oscillation after this recovery.

1. `STP_RUN` is bit [4] of `HWWDE` register. It enables running the Hardware Watchdog in STOP mode. `STP_RUN` can only be set by software, but not cleared. `STP_RUN` is cleared by INIT.

WARNING: If a Hardware Watchdog timeout INIT signal appears just at the transition to Standby Mode, the device may enter an unpredictable state. Always make sure that the hardware Watchdog has been cleared just before entering Shutdown.

■ Clock Supervisor in Shutdown

The INITX pin, Clock Supervisor and Hardware Watchdog form the “external INIT chain”, like shown in the figure in section [“Determining the Reset Source after Shutdown” on page 1642](#). The Shutdown control is part of this chain.

An INIT signal from the Clock Supervisor will pass the Hardware Watchdog and arrive at the same Shutdown control input line as the INIT signal from Hardware Watchdog. Therefore, clock supervision in Shutdown mode is only possible if the Hardware Watchdog is operating in parallel.

If the Hardware Watchdog is disabled in Shutdown mode, an INIT signal from the Clock Supervisor is ignored in Shutdown.

The Clock Supervisor is enabled by default. In Shutdown mode, as long as the Main- and/or Sub-oscillator is running and the RC clock is not stopped, the CSV is supervising the Main- or Sub-oscillator, respectively.

- The Clock Supervisor needs the RC clock, so set CSVCR.RCE=1, this is default setting.
- If the Main-oscillator is not stopped (STCR.OSCD1=0), the Main clock supervisor is running.
- If the Main-oscillator fails, the Main Clock Supervisor generates INIT, which can cancel the Shutdown state and force recovery. The CPU runs on RC clock during and after the recovery.
- If the Sub-oscillator is not stopped (STCR.OSCD2=0), the Sub Clock Supervisor is running.
- If the Sub-oscillator fails, the Sub clock is switched to RC clock divided by 2. An INIT is not generated, and the Real Time Clock continues running on RC clock divided by 2, if RTC is enabled.

To disable the Clock Supervisor, clear the bits CSVCR.MSVE and CSVCR.SSVE.

■ Low Voltage Detection in Shutdown

Low Voltage Detection is not supported in Shutdown mode. Always set the Low Voltage Detection into power down mode (LVDET.LVEPD=1, LVDET.LVIPD=1) before enabling Shutdown.

■ External Interrupts: Input Voltage Setting

The input voltages (CMOS-Schmitt, Automotive, TTL, CMOS-2) of the external interrupt lines are defined by the setting of PILR and EPILR of the appropriate ports. The PILR and EPILR settings for the 8 external recovery interrupt lines are maintained during Shutdown mode until they are cleared by the Software reset which follows the Recovery INIT. [Table 64.4-1 on page 1641](#) shows the PILR/EPILR register bits:

MB91460 Series**Table 64.4-1 Port Input Level Registers of the external shutdown recovery interrupt pins**

EPILR	PILR	Pin No ¹	Pin Name
EPILR23[2]	PILR23[2]	93	P23_2/RX1/INT9
EPILR23[0]	PILR23[0]	91	P23_0/RX0/INT8
EPILR24[7]	PILR24[7]	90	P24_7/SCL3/INT7C
EPILR24[6]	PILR24[6]	89	P24_6/SDA3/INT6D
EPILR24[3]	PILR24[3]	86	P24_3/INT3
EPILR24[2]	PILR24[2]	85	P24_2/INT2
EPILR24[1]	PILR24[1]	84	P24_1/INT1
EPILR24[0]	PILR24[0]	83	P24_0/INT0

1. The pin numbers of MB91F467E are listed here.

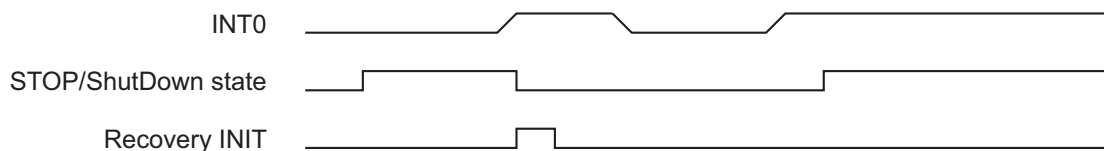
■ External Interrupts: Level or Edge Setting

The registers EXTLV1 and EXTLV2 are used to set the interrupt level or edge for recovery per interrupt channel.

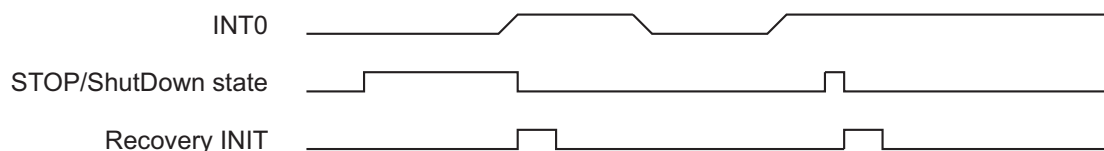
LBx	LAx	Interrupt Level
0	0	"L" level (initial value)
0	1	"H" level
1	0	Rising edge
1	1	Falling edge

The settings “level” and “edge” generate different behaviour if the external source line is not changed back after recovery of if it changes to the sensitive level before Shutdown. Examples:

- INT0 is enabled for recovery on **rising edge**. If a rising edge appears during Shutdown state, recovery is performed. If a rising edge is outside Shutdown state, there will be no recovery:



- INT0 is enabled for recovery on **high level**. If INT0 changes to high level during Shutdown state, recovery is performed. If INT0 changes to high level already before Shutdown state, the Shutdown is recovered immediately because the high level on INT0 is valid. Note that, in this case, a complete shut-down/power-up sequence with recovery INIT is performed:



Note: If “H” level or “L” level is enabled for recovery, the level must be active for minimum 500 μs.

64.4.2 Recovery from shutdown mode

The following factors are available to recover from the shutdown state:

- Assert the reset signal at the INITX terminal for minimum 10 ms ¹
- Input of a valid recovery request via an external interrupt terminal
- Real Time Clock Interrupt (when RTC interrupt is enabled)
- Hardware Watchdog reset (when HWWD is enabled in STOP mode)
- Main Clock Supervisor reset (when Main oscillator is running and Main Clock Supervisor is enabled and recovery by HWWD is enabled)

Shutdown state is released when a valid recovery factor is permitted. After the Shutdown state release, the device restarts with a settings initialization reset (INIT), just like power-up operation. Only the Real Time Clock, the Oscillation Stabilization settings in STCR register, and the recovery source flags in the Shutdown registers EXTF and SHDINT are not cleared.

The internal restart sequence is as follows:

1. Resume the internal power supply.
2. Reset and assert the initialization reset (INIT).
3. Wait for oscillation stabilization.
4. Start the reset sequence.

As the external interrupt source flags and the RTC flag are retained in EXTF and SHDINT registers, it is possible to determine whether it is power-up operation or recovery from shutdown state by checking the flags.

■ The Real Time Clock at Recovery from Shutdown

In normal operation, the registers and settings of the Real Time Clock are initialized by Software Reset (RST).

At recovery from Shutdown, the RTC is **not** initialized:

- The prescaler, second, minute and hour counters continue counting also during the recovery INIT state.
- The clock selection for the RTC (by CSCFG.OSC1, CSCFG.OSC0) remains unchanged.
- The RTC interrupt enable bits and interrupt flags (in WTCR and WTCE registers) remain unchanged.

So at each recovery from Shutdown, the RTC continues running and the current time as well as the interrupt flags can be read from the RTC after recovery.

Note: The Interrupt Control Register for RTC (ICR58), the Interrupt Level Mask (ILM) register as well as the Condition Code Register (CCR, containing the I-Flag) are cleared by the recovery INIT, so that all interrupt processing is disabled.

If the software re-enables interrupt processing by setting ICR58, ILM and I-Flag, the software will process the pending RTC interrupt immediately.

64.4.3 Determining the Reset Source after Shutdown

The recovery from Shutdown is followed by an Setting Initialization Reset (INIT). Because INIT is always followed by a Software Reset (RST), the CPU fetches the Mode- and Reset-Vectors and jumps to the Reset Vector, which is located in the Boot ROM.

Figure 64.4-1 on page 1643 shows how the Shutdown Control is located in the external INIT chain:

1. The minimum INTX=0 pulse length is determined by the time the main oscillator needs for stabilization.

Figure 64.4-1 Shutdown Control in the external INIT chain:

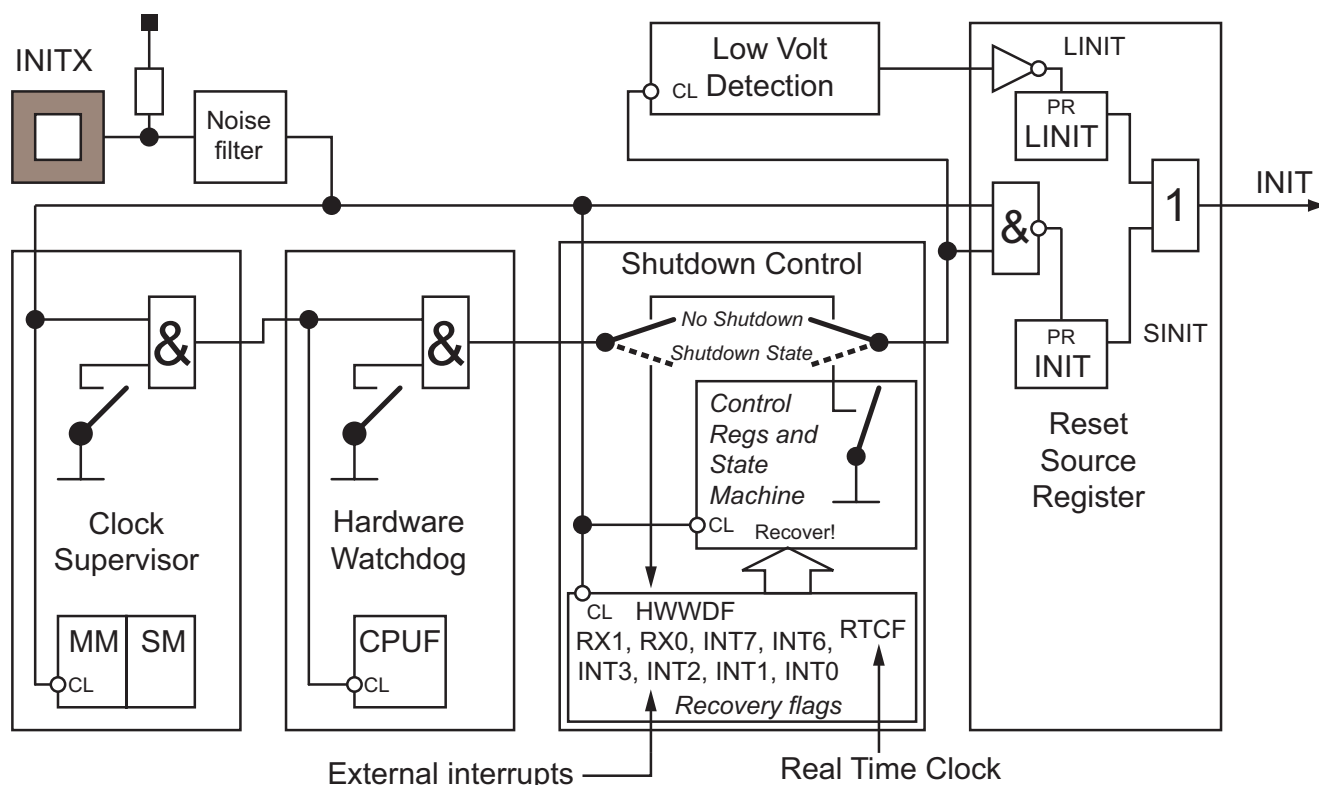


Table 64.4-2 lists the registers and flags for determination of the reset source, including Shutdown:

Table 64.4-2 Registers and flags for determination of the reset source

Register	Addr.	7	6	5	4	3	2	1	0
RSRR	480 _H ¹	INIT	HSTB	WDOG	ERST	SRST	LINIT	WT1	WT0
EXTF	4D7 _H	RX1	RX0	INT7	INT6	INT3	INT2	INT1	INT0
SHDINT	4DB _H	-	-	-	-	HWWDF	HWWDE	RTCF	RTCE
CSVCR	4AD _H	SCKS	MM	SM	RCE	MSVE	SSVE	SRST	OUTE
HWWD	4C7 _H	-	-	-	-	CL	-	-	CPUF

1. RSRR is read and cleared by the Boot ROM software. After Boot ROM, the content of RSRR can be found in CPU register R4[7:0] and in a variable in memory.
Please refer to section 53.3.1 Reset Source Register (RSRR) (Page No.1170)

Note: RSRR: Reset Source register, see 9.4.1 RSRR: Reset Cause Register (Page No.240)

EXTF: External shutdown recovery flags, see page 1636

SHDINT: Hardware Watchdog/ Real Time Clock recovery flags, see page 1634

CSVCR: CLock Supervisor Control / Status register, see 16.2. Clock Supervisor Register (Page No.335)

HWWD: Hardware Watchdog register, see 21.2.1 Hardware watchdog timer control and status register (Page No.401)

Recovery from Shutdown will set the INIT bit in RSRR register. Because the INIT bit can also be set by external INIT (low level at INITX pin), Clock Supervisor or Hardware Watchdog, the flags in EXTF, SHDINT, CSVCR and HWWD should be checked for determining the reset source.

The recovery flags in EXTf and SHDINT are set **only** in Shutdown mode and **only** if recovery by this channel is enabled.

64.4.4 Registers which are not initialized by Shutdown Recovery

As described above, recovery from Shutdown performs a settings initialization reset (INIT) followed by software reset (RST). This sequence will initialize the complete device with some exceptions, explained in the following table.

Table 64.4-3 Registers which are not initialized by Shutdown Recovery:

Register	Address	non-initialized Bits	Reason
STCR	481 _H	OS1, OS0	Keep oscillation stabilization time setting
CSVCR	4AD _H	all bits	Clock Supervisor is not initialized by recovery
CSCFG	4AE _H	all bits	Keep RTC and Calibration clock source settings
CMCFG	4AF _H	all bits	Keep Clock Monitor settings
WTCER	4A1 _H	all bits	Real Time Clock to continue running
WTCR	4A2 _H - 4A3 _H		
WTBR	4A5 _H - 4A7 _H		
WTHR	4A8 _H		
WTMR	4A9 _H		
WTSR	4AA _H		
CUCR	4B0 _H - 4B1 _H	all bits	Subclock Calibration unit is part of RTC module
CUTD	4B2 _H - 4B3 _H		
CUTR1	4B4 _H - 4B5 _H		
CUTR2	4B6 _H - 4B7 _H		
HWWE	4C6 _H	all bits	Hardware Watchdog is not initialized by recovery
HWWD	4C7 _H	all bits	
EXTF	4D7 _H	all bits	Keep external recovery flags
SHDINT	4DB _H	HWWDf, RTCF	Keep hardware watchdog and RTC recovery flags

MB91460 Series

64.4.5 I/O Behaviour in Shutdown

During Shutdown mode, the MB91F467E I/O pins are switched into dedicated states:

Table 64.4-4 Pin states in ShutDown mode

Ports/Pins	Port function		Setting
P00_0 to P00_7, P01_0 to P01_7, P02_0 to P02_7, P03_0 to P03_7.	D[31:0]	External bus data I/O	The pins are switched to input direction, but it is not possible to input signals on these pins.
P08_6, P08_7, P10_5, P13_0	BRQ, RDY, MCLKI, DREQ0	External bus control inputs	If STCR.HIZ (HiZ mode in STOP) is not set, the pull-up/pull-down settings are maintained during shutdown. If STCR.HIZ is set, the pull-up and pull-down resistors are disabled.
P04_0 to P04_1, P05_0 to P05_7, P06_0 to P06_7, P07_0 to P07_7.	A[25:0]	External bus address outputs	
P08_0 to P08_5, P09_0 to P09_7, P10_1 to P10_4, P10_6, P13_1, P13_2	WRnX, RDX, BGRNTX, CSnX, ASX, BAAX, WEX, MCLKO, MCLKE	External bus control and clock outputs	If the pins were switched to output direction before shutdown (by PFR==1 or DDR==1), the pins will output '1' value and the driver strength is switched to 2 mA. Otherwise, the pins keep input direction, but it is not possible to input signals on these pins. If STCR.HIZ is not set, the pull-up/pull-down settings are maintained. If STCR.HIZ is set, the pull-up and pull-down resistors are disabled.
P24_0 to P24_3, P24_6, P24_7, P23_0, P23_2	INT0 to INT3, INT6, INT7, RX0/INT8, RX1/INT9	Pins used for Shutdown recovery	The pins are switched to input direction. If STCR.HIZ is not set, the pull-up/pull-down settings are maintained during shutdown. If STCR.HIZ is set, the pull-up and pull-down resistors are disabled. If external interrupt is enabled for recovery from Shutdown (Shutdown INTE=1), the input threshold setting (PILR, EPILR) is maintained during the shutdown mode and it is possible to input signals for recovery. After the first recovery factor is accepted, the port settings are initialized when the device proceeds to the reset (INIT/RST) sequence.
Pnn_m ¹	all other Ports not mentioned above		All other pins are switched to input direction, but it is not possible to input signals on these pins. If STCR.HIZ is not set, the pull-up/pull-down settings are maintained during Shutdown. If STCR.HIZ is set, the pull-up and pull-down resistors are disabled.
ALARM_0	ALARM analog input		The state of ALARM input is not changed in Shutdown state.
MD_0 to MD_2	Mode inputs		The state of MD[2:0] is not changed in Shutdown state
INITX	External INIT		The state of INITX is not changed in Shutdown state. The pull-up is enabled. It is possible to input external INITX signal during Shutdown.
VCC18C	Regulator capacitor pin		The capacitor connection pin for internal regulator shows the voltage which is applied to internal Always-ON domain.

1. nn = 14 to 29, m = 0 to 7

MB91460 Series**Chapter 65 Electrical Specification**

See the corresponding data sheet for the electrical specification of each device.

MB91460 Series

Appendix A Instruction Lists

This section includes Instruction Lists of MB91460 super-series CPU.

[A.1 Meaning of Symbols](#)

[A.2 Instruction Lists](#)

[A.3 Instruction Maps](#)

[A.4 Instruction Maps of Instruction Format TYPE-E](#)

A.1 Meaning of Symbols

This section describes the meaning of symbols used in the Instruction Lists.

A.1.1 Mnemonic and Operation Columns

These are the symbols used in Mnemonic and Operation columns of Instruction Lists.

i4

It is 4-bit immediate data. 0(0_H) to 15(F_H) in case of zero extension and -16(0_H) to -1(F_H) in case of minus extension can be specified.

Table A.1-1 Zero Extension and Minus Extension Values of 4-bit Immediate Data

Bit Pattern	Specified Value	
	Zero Extension	Minus Extension
0000 _B	0	-16
0001 _B	1	-15
0010 _B	2	-14
...		
1101 _B	13	-3
1110 _B	14	-2
1111 _B	15	-1

i8

8-bit immediate data, range 0 (00_H) to 255 (FF_H)

i20

20-bit immediate data, range 0 (00000_H) to 1,048,575 (FFFFF_H)

i32

32-bit immediate data, range 0 (0000 0000_H) to 4,294,967,295 (FFFF FFFF_H)

s8

Signed 8-bit immediate data, range -128 (80_H) to 127 (7F_H)

s10

Signed 10-bit immediate data, range -512 (200_H) to 508 (1FC_H) in multiples of 4

u4

Unsigned 4-bit immediate data, range 0 (0_H) to 15 (F_H)

MB91460 Series**u8**Unsigned 8-bit immediate data, range 0 (00_H) to 255 (FF_H)**u10**Unsigned 10-bit immediate data, range 0 (000_H) to 1020 (3FC_H) in multiples of 4**udisp6**Unsigned 6-bit address values, range 0 (00_H) to 60 (3C_H) in multiples of 4**disp8**Signed 6-bit address values, range -128(80_H) to 127(7F_H)**disp9**Signed 9-bit address values, range -256(100_H) to 254(0FE_H) in multiples of 2**disp10**Signed 10-bit address values, range -512(200_H) to 508(1FC_H) in multiples of 4**dir8**Unsigned 8-bit address values, range 0 (00_H) to 255 (FF_H)**dir9**Unsigned 9-bit address values, range 0 (000_H) to 510 (1FE_H) in multiples of 2**dir10**Unsigned 10-bit address values, range 0 (000_H) to 1020 (3FC_H) in multiples of 4**label9**Branch address, range 256 (100_H) to 254 (0FE_H) in multiples of 2 for the value of Program Counter (PC) +2**label12**Branch address, range - 2048 (800_H) to 2046 (7FE_H) in multiples of 2 for the value of Program Counter (PC) +2**rel8**Signed 8-bit relative address. Result which is double the value of rel8 for the value of Program Counter (PC) +2 will denote the Branch Destination Address. Range -128 (80_H) to 127 (7F_H)**rel11**Signed 11-bit relative address. Result which is double the value of rel11 for the value of Program Counter (PC) +2 will denote the Branch Destination Address. Range -1024 (400_H) to 1023 (3FF_H)

Ri, Rj

Indicates General-purpose Registers (R0 to R15)

Table A.1-2 Specification of General-purpose Register based on Rj/Ri

Ri / Rj	Register	Ri / Rj	Register
0000	R0	1000	R8
0001	R1	1001	R9
0010	R2	1010	R10
0011	R3	1011	R11
0100	R4	1100	R12
0101	R5	1101	R13
0110	R6	1110	R14
0111	R7	1111	R15

Rs

Indicates Dedicated Registers (TBR, RP, USP, SSP, MDH, MDL)

Table A.1-3 Specification of Dedicated Register based on Rs

Rs	Register	Rs	Register
0000	Table Base Register (TBR)	1000	Reserved (Disabled)
0001	Return Pointer (RP)	1001	
0010	System Stack Pointer (SSP)	1010	
0011	User Stack Pointer (USP)	1011	
0100	Multiplication/Division Register (MDH)	1100	
0101	Multiplication/Division Register (MDL)	1101	
0110	Reserved (Disabled)	1110	
0111		1111	

MB91460 Series

(reglist)

Indicates 8-bit Register list. Register corresponding to each bit value can be specified.

Table A.1-4 Correspondence between reglist of LDM0, LDM1 Instruction and General-purpose Register

LDM0 Instruction		LDM1 Instruction	
reglist	Register	reglist	Register
bit0	R0	bit0	R8
bit1	R1	bit1	R9
bit2	R2	bit2	R10
bit3	R3	bit3	R11
bit4	R4	bit4	R12
bit5	R5	bit5	R13
bit6	R6	bit6	R14
bit7	R7	bit7	R15

Table A.1-5 Correspondence between reglist of STM0, STM1 Instruction and General-purpose Register

STM0 Instruction		STM1 Instruction	
reglist	Register	reglist	Register
bit0	R7	bit0	R15
bit1	R6	bit1	R14
bit2	R5	bit2	R13
bit3	R4	bit3	R12
bit4	R3	bit4	R11
bit5	R2	bit5	R10
bit6	R1	bit6	R9
bit7	R0	bit7	R8

A.1.2 Operation Column

These are symbols used in Operation Column of Instruction Lists and operation of Detailed Execution Instructions.

`extu()`

Indicates a zero extension operation, in which portion lacking higher bits is complemented by adding "0" bit.

`extn()`

Indicates a minus extension operation, in which portion lacking higher bits is complemented by adding "1" bit.

`exts()`

Indicates a sign extension operation, in which zero extension is performed for the data within () if MSB is "0" and a minus extension is performed if MSB is "1".

`&`

Indicates logical calculation of each bit (AND)

`|`

Indicates the logical sum of each bit (OR)

`^`

Indicates Dedicated Logical Sum of each bit (EXOR)

`()`

Indicates specification of indirect address. It is address memory read/write value of the Register or formula within ().

`{ }`

Indicate the calculation priority. Since () is used for specifying indirect address, different bracket namely { } is used.

`if (Condition) then {formula} or if (condition) then {Formula 1} else {Formula 2}`

Indicates the execution of conditions. If the conditions are established, formula after 'then' is executed and when the conditions are not established, formula next to 'else' is executed. Formula can be described variously using the { }.

`[m:n]`

Bits from m to n are extracted and targeted for operation.

MB91460 Series

A.1.3 Format Column

Symbols used in the Format Column of the Instruction Lists.

A to H

Indicates the Instruction Formats. A to H correspond to TYPE-A to TYPE-H.

A.1.4 OP Column

Hexadecimal value used in the Instruction Lists. They denote operation codes (OP, SUB-OP). They branch into the following depending on the Instruction Format.

TYPE-A, TYPE-C, TYPE-D, TYPE-G

2-digit hexadecimal value represents 8-bit OP code

TYPE-B

2-digit hexadecimal value represents higher 4 bits of OP code with lower 4 bits of 0000_B.

TYPE-E, TYPE-E', TYPE-H

4-digit hexadecimal value represents higher 8 bits of OP code with higher 2 digits, 4-bits of SUB-OP code with the next 1 digit and the remainder with "0".

TYPE-F

2-digit hexadecimal code represents higher 5 bits of OP code with lower 3 bits of 000_B.

A.1.5 CYC Column

Symbols used in CYC Column of Instruction Lists and the member of execution cycles of Detailed Execution Instructions. Numerical values represent CPU clock cycles.

a

Memory access cycles. Cycles change depending on the access target. Minimum value is 1 cycle.

b

Memory access cycles. Cycles change depending on the access target. Minimum value is 1 cycle.

When the Register which is target of load operation is referred to by the succeeding Instruction, an interlock will be applied from that point and the number of execution cycles will increase by 1.

c

An interlock will be applied when the immediately next Instruction is read and written to Multiplication/Division Register (R15, SSP and USP).

An interlock will be applied when the immediately next Instruction is the Instruction Format A.

Cycles will be increased by 1. Otherwise it will be 2 cycles. However, minimum value is 1 cycle.

d

An interlock will be applied when the immediately next Instruction refers to Multiplication/Division Register (MDH/MDL) and the number of execution cycles will be increased by 1. Otherwise it will be 2 cycles. However, Minimum value is 1 cycle.

An interlock will be always applied when the Special Register (TBR, RP, USP, SSP, MDH, or MDL) is accessed by the "ST Rs, @R15-" instruction located immediately after the DIV1 Instruction. The number of

execution cycles will be increased by 1 with the interlock. Otherwise it will be 2 cycles.

A.1.6 FLAG Column

Symbols used for flag change in the Flag Column of Instruction Lists and Detailed Execution Instructions. Represents change in Negative Flag (N), Zero Flag (Z), Overflow Flag (V), Carry Flag (C) of the Condition Code Register (CCR).

C

Varies depending on the result of operation

-

No change

0

Value becomes "0"

1

Value becomes "1"

A.1.7 RMW Column

Symbols used in the RMW Column of Instruction Lists. It represents whether or not it is Read-Modify-Write Instruction.

-

Instruction is not Read-Modify-Write Instruction.

m

Instruction is Read-Modify-Write Instruction.

MB91460 Series

A.2 Instruction Lists

This section indicates Instruction Lists of MB91460 Super-series CPU.

There are a total of 165 instructions in MB91460 Super-series CPU. These instructions are divided into the following 16 categories.

- Add/Subtract Instructions (10Instructions)
- Compare Calculation Instructions (3 Instructions)
- Logical Calculation Instructions (12 Instructions)
- Bit Operation Instructions (8 Instructions)
- Multiply/ Divide Instructions (10 Instructions)
- Shift Instructions (9 Instructions)
- Immediate Data Transfer Instructions (3 Instructions)
- Memory Load Instructions (13 Instructions)
- Memory Store Instructions (13 Instructions)
- Inter-Register Transfer Instructions/Dedicated Register Transfer Instructions (5 Instructions)
- Non-delayed Branching Instructions (23 Instructions)
- Delayed Branching Instructions (20 Instructions)
- Direct Addressing Instructions (14 Instructions)
- Other Instructions (16 Instructions)
- Resource instruction (2 Instructions)
- Coprocessor control instruction (4 Instructions)

Table A.2-1 Add/Subtract Instructions (10 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	-	Ri+Rj → Ri	
ADD #i4, Ri	C	A4	1	CCCC	-	Ri+extu(i4) → Ri	i4 is zero extension
ADD2 #i4, Ri	C	A5	1	CCCC	-	Ri+extn(i4) → Ri	i4 is Minus extension
ADDC Rj, Ri	A	A7	1	CCCC	-	Ri+Rj+C → Ri	Add with carry
ADDN Rj, Ri	A	A2	1	----	-	Ri+Rj → Ri	
ADDN #i4, Ri	C	A0	1	----	-	Ri+extu(i4) → Ri	i4 is Zero extension
ADDN2 #i4, Ri	C	A1	1	----	-	Ri+extn(i4) → Ri	i4 is Minus extension
SUB Rj, Ri	A	AC	1	CCCC	-	Ri-Rj → Ri	
SUBC Rj, Ri	A	AD	1	CCCC	-	Ri-Rj-C → Ri	Add with carry
SUBN Rj, Ri	A	AE	1	----	-	Ri-Rj → Ri	

Table A.2-2 Compare Calculation Instructions (3 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	-	Ri-Rj	
CMP #i4, Ri	C	A8	1	CCCC	-	Ri-extu(i4)	i4 is Zero extension
CMP2 #i4, Ri	C	A9	1	CCCC	-	Ri-extn(i4)	i4 is Minus extension

Table A.2-3 Logical Calculation Instructions (12 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
AND Rj, Ri	A	82	1	CC--	-	Ri & Rj → Ri	Word
AND Rj, @Ri	A	84	1+2a	CC--	m	(Ri) & Rj → (Ri)	Word
ANDH Rj, @Ri	A	85	1+2a	CC--	m	(Ri) & Rj → (Ri)	Half-Word
ANDB Rj, @Ri	A	86	1+2a	CC--	m	(Ri) & Rj → (Ri)	Byte
OR Rj, Ri	A	92	1	CC--	-	Ri Rj → Ri	Word
OR Rj, @Ri	A	94	1+2a	CC--	m	(Ri) Rj → (Ri)	Word
ORH Rj, @Ri	A	95	1+2a	CC--	m	(Ri) Rj → (Ri)	Half-Word
ORB Rj, @Ri	A	96	1+2a	CC--	m	(Ri) Rj → (Ri)	Byte
EOR Rj, Ri	A	9A	1	CC--	-	Ri ^ Rj → Ri	Word
EOR Rj, @Ri	A	9C	1+2a	CC--	m	(Ri) ^ Rj → (Ri)	Word
EORH Rj, @Ri	A	9D	1+2a	CC--	m	(Ri) ^ Rj → (Ri)	Half-Word
EORB Rj, @Ri	A	9E	1+2a	CC--	m	(Ri) ^ Rj → (Ri)	Byte

Table A.2-4 Bit Operation Instructions (8 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
BANDL #u4, @Ri	C	80	1+2a	----	m	(Ri) & {F0 _H +u4} → (Ri)	Lower 4- bit
BANDH #u4, @Ri	C	81	1+2a	----	m	(Ri) & {u4<<4+0F _H } → (Ri)	Higher 4 bit
BORL #u4, @Ri	C	90	1+2a	----	m	(Ri) u4 → (Ri)	Lower 4- bit
BORH #u4, @Ri	C	91	1+2a	----	m	(Ri) {u4<<4} → (Ri)	Higher 4 bit
BEORL #u4, @Ri	C	98	1+2a	----	m	(Ri) ^ u4 → (Ri)	Lower 4- bit
BEORH #u4, @Ri	C	99	1+2a	----	m	(Ri) ^ {u4<<4} → (Ri)	Higher 4 bit
BTSTL #u4, @Ri	C	88	2+a	0C--	-	(Ri) & u4	Lower 4- bit
BTSTH #u4, @Ri	C	89	2+a	CC--	-	(Ri) & {u4<<4}	Higher 4 bit

MB91460 Series

Table A.2-5 Multiply/ Divide Instructions (10 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MUL Rj, Ri	A	AF	5	CCC-	-	$Ri \times Rj \rightarrow MDH, MDL$	$32 \times 32 \text{ bit} = 64 \text{ bit}$
MULU Rj, Ri	A	AB	5	CCC-	-	$Ri \times Rj \rightarrow MDH, MDL$	Unsigned
MULH Rj, Ri	A	BF	3	CC--	-	$Ri \times Rj \rightarrow MDL$	$16 \times 16 \text{ bit} = 32 \text{ bit}$
MULUH Rj, Ri	A	BB	3	CC--	-	$Ri \times Rj \rightarrow MDL$	Unsigned
DIV0S Ri	E	97-4	1	----	-	In the Specified Instruction Sequence $MDL \div Ri \rightarrow MDL$ $MDL \% Ri \rightarrow MDH$	Step Calculation $32 \div 32 \text{ bit} = 32 \text{ bit}$
DIV0U Ri	E	97-5	1	----	-		
DIV1 Ri	E	97-6	d	-C-C	-		
DIV2 Ri	E	97-7	1	-C-C	-		
DIV3	E'	9F-6	1	----	-		
DIV4S	E'	9F-7	1	----	-		

Table A.2-6 Shift Instructions (9 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	-	$Ri \ll Rj \rightarrow Ri$	Logical Shift
LSL #u4, Ri	C	B4	1	CC-C	-	$Ri \ll u4 \rightarrow Ri$	
LSL2 #u4, Ri	C	B5	1	CC-C	-	$Ri \ll \{u4+16\} \rightarrow Ri$	
LSR Rj, Ri	A	B2	1	CC-C	-	$Ri \gg Rj \rightarrow Ri$	Logical Shift
LSR #u4, Ri	C	B0	1	CC-C	-	$Ri \gg u4 \rightarrow Ri$	
LSR2 #u4, Ri	C	B1	1	CC-C	-	$Ri \gg \{u4+16\} \rightarrow Ri$	
ASR Rj, Ri	A	BA	1	CC-C	-	$Ri \gg Rj \rightarrow Ri$	Arithmetic Shift
ASR #u4, Ri	C	B8	1	CC-C	-	$Ri \gg u4 \rightarrow Ri$	
ASR2 #u4, Ri	C	B9	1	CC-C	-	$Ri \gg \{u4+16\} \rightarrow Ri$	

Table A.2-7 Immediate Data Transfer Instructions (3 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LDI:32 #i32, Ri	H	9F-8	3	----	-	$i32 \rightarrow Ri$	
LDI:20 #i20, Ri	G	9B	2	----	-	$extu(i20) \rightarrow Ri$	Higher 12-Bits are Zero extension
LDI:8 #i8, Ri	B	C0	1	----	-	$extu(i8) \rightarrow Ri$	Higher 24-Bits are Zero extension

Table A.2-8 Memory Load Instructions (13 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LD @Rj, Ri	A	04	b	----	-	(Rj) → Ri	Word
LD @(R13, Rj), Ri	A	00	b	----	-	(R13+Rj) → Ri	
LD @(R14, disp10), Ri	B	20	b	----	-	(R14+o8 × 4) → Ri	
LD @(R15, udisp6), Ri	C	03	b	----	-	(R15+u4 × 4) → Ri	
LD @R15+, Ri	E	07-0	b	----	-	(R15) → Ri, R15+4 → R15	
LD @R15+, Rs	E	07-8	b	----	-	(R15) → Rs, R15+4 → R15	
LD @R15+, PS	E	07-9	1+a+c	CCCC	-	(R15) → PS, R15+4 → R15	Half- Word Zero extension
LDUH @Rj, Ri	A	05	b	----	-	extu((Rj)) → Ri	
LDUH @(R13, Rj), Ri	A	01	b	----	-	extu((R13+Rj)) → Ri	
LDUH @(R14, disp9), Ri	B	40	b	----	-	extu((R14+o8 × 2)) → Ri	
LDUB @Rj, Ri	A	06	b	----	-	extu((Rj)) → Ri	
LDUB @(R13, Rj), Ri	A	02	b	----	-	extu((R13+Rj)) → Ri	
LDUB @(R14, disp8), Ri	B	60	b	----	-	extu((R14+o8)) → Ri	Byte Zero extension

- Relation of field "o8" in the Instruction Format TYPE-B and field "u4" in TYPE-C Format to the values disp8 to disp10, udisp6 in assembly notation is as follows.

o8 = disp8

o8 = disp9 >> 1

o8 = disp10 >> 2

u4 = udisp6 >> 2

MB91460 Series**Table A.2-9 Memory Store Instructions (13 Instructions)**

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ST Ri, @Rj	A	14	a	----	-	Ri → (Rj)	Word
ST Ri, @(R13, Rj)	A	10	a	----	-	Ri → (R13+Rj)	
ST Ri, @(R14, disp10)	B	30	a	----	-	Ri → (R14+o8 × 4)	
ST Ri, @(R15, udisp6)	C	13	a	----	-	Ri → (R15+u4 × 4)	
ST Ri, @-R15	E	17-0	a	----	-	R15-4 → R15, Ri → (R15)	
ST Rs, @-R15	E	17-8	a	----	-	R15-4 → R15, Rs → (R15)	
ST PS, @-R15	E	17-9	a	----	-	R15-4 → R15, PS → (R15)	Half- Word
STH Ri, @Rj	A	15	a	----	-	Ri → (Rj)	
STH Ri, @(R13, Rj)	A	11	a	----	-	Ri → (R13+Rj)	
STH Ri, @(R14, disp9)	B	50	a	----	-	Ri → (R14+o8 × 2)	Byte
STB Ri, @Rj	A	16	a	----	-	Ri → (Rj)	
STB Ri, @(R13, Rj)	A	12	a	----	-	Ri → (R13+Rj)	
STB Ri, @(R14, disp8)	B	70	a	----	-	Ri → (R14+o8)	

- Relation of field "o8" in the Instruction Format TYPE-B and field "u4" in TYPE-C Format to the values disp8 to disp10, udisp6 in assembly notation is as follows.

o8 = disp8

o8 = disp9 >> 1

o8 = disp10 >> 2

u4 = udisp6 >> 2

Table A.2-10 Inter-Register Transfer Instructions/Dedicated Register Transfer Instructions (5 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	-	Rj → Ri	Transfer between general-purpose Registers
MOV Rs, Ri	A	B7	1	----	-	Rs → Ri	Rs: Dedicated Register
MOV Ri, Rs	A	B3	1	----	-	Ri → Rs	Rs: Dedicated Register
MOV PS, Ri	E	17-1	1	----	-	PS → Ri	PS: Program Status
MOV Ri, PS	E	07-1	c	CCCC	-	Ri → PS	PS: Program Status

Table A.2-11 Non-delayed Branching Instructions (23 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
JMP @Ri	E	97-0	2	----	-	Ri → PC	
CALL label12	F	D0	2	----	-	PC+2 → RP, PC+2+exts(rel11 × 2) → PC	
CALL @Ri	E	97-1	2	----	-	PC+2 → RP, Ri → PC	
RET	E'	97-2	2	----	-	RP → PC	
INT #u8	D	1F	3+3a	----	-	SSP-4 → SSP, PS → (SSP), SSP-4 → SSP, PC+2 → (SSP), 0 → CCR:I, 0 → CCR:S, (TBR+3FC-u8 × 4) → PC	
INTE	E'	9F-3	3+3a	----	-	SSP → SSP, PS → (SSP), SSP → SSP, PC+2 → (SSP), 0 → CCR:S, 4 → ILM, (TBR+3D8) → PC	
RETI	E'	97-3	2+2a	----	-	(SSP) → PC, SSP+4 → SSP, (SSP) → PS, SSP+4 → SSP	
BNO label9	D	E1	1	----	-	No branch	
BRA label9	D	E0	2	----	-	PC+2+exts(rel8 × 2) → PC	
BEQ label9	D	E2	2/1	----	-	if (Z==1) then PC+2+exts(rel8 × 2) → PC	
BNE label9	D	E3	2/1	----	-	if (Z==0) then PC+2+exts(rel8 × 2) → PC	
BC label9	D	E4	2/1	----	-	if (C==1) then PC+2+exts(rel8 × 2) → PC	
BNC label9	D	E5	2/1	----	-	if (C==0) then PC+2+exts(rel8 × 2) → PC	
BN label9	D	E6	2/1	----	-	if (N==1) then PC+2+exts(rel8 × 2) → PC	
BP label9	D	E7	2/1	----	-	if (N==0) then PC+2+exts(rel8 × 2) → PC	
BV label9	D	E8	2/1	----	-	if (V==1) then PC+2+exts(rel8 × 2) → PC	
BNV label9	D	E9	2/1	----	-	if (V==0) then PC+2+exts(rel8 × 2) → PC	
BLT label9	D	EA	2/1	----	-	if (V ^ N==1) then PC+2+exts(rel8 × 2) → PC	
BGE label9	D	EB	2/1	----	-	if (V ^ N==0) then PC+2+exts(rel8 × 2) → PC	
BLE label9	D	EC	2/1	----	-	if ({V ^ N} Z==1) then PC+2+exts(rel8 × 2) → PC	
BGT label9	D	ED	2/1	----	-	if ({V ^ N} Z==0) then PC+2+exts(rel8 × 2) → PC	
BLS label9	D	EE	2/1	----	-	if (C or Z==1) then PC+2+exts(rel8 × 2) → PC	
BHI label9	D	EF	2/1	----	-	if (C or Z==0) then PC+2+exts(rel8 × 2) → PC	

- The value of "2/1" in CYC Column indicates 2 cycles if branching and 1 if not branching.
- It is necessary to set the Stack Flag (S) to "0" for RETI execution.
- The field "rel8" in TYPE_D Instruction Format and the field "rel11" in TYPE-F Format have the following relation to the values of label9, label12 in assembly notation.

$$\text{rel8} = (\text{label9} - \text{PC} - 2) / 2$$

$$\text{rel11} = (\text{label12} - \text{PC} - 2) / 2$$

MB91460 Series**Table A.2-12 Delayed Branching Instructions (20 Instructions)**

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
JMP:D @Ri	E	9F-0	1	----	-	Ri → PC	
CALL:D label12	F	D8	1	----	-	PC+4 → RP, PC+2+exts(rel11 × 2) → PC	
CALL:D @Ri	E	9F-1	1	----	-	PC+4 → RP, Ri → PC	
RET:D	E'	9F-2	1	----	-	RP → PC	
BNO:D label9	D	F1	1	----	-	No branch	
BRA:D label9	D	F0	1	----	-	PC+2+exts(rel8 × 2) → PC	
BEQ:D label9	D	F2	1	----	-	if (Z==1) then PC+2+exts(rel8 × 2) → PC	
BNE:D label9	D	F3	1	----	-	if (Z==0) then PC+2+exts(rel8 × 2) → PC	
BC:D label9	D	F4	1	----	-	if (C==1) then PC+2+exts(rel8 × 2) → PC	
BNC:D label9	D	F5	1	----	-	if (C==0) then PC+2+exts(rel8 × 2) → PC	
BN:D label9	D	F6	1	----	-	if (N==1) then PC+2+exts(rel8 × 2) → PC	
BP:D label9	D	F7	1	----	-	if (N==0) then PC+2+exts(rel8 × 2) → PC	
BV:D label9	D	F8	1	----	-	if (V==1) then PC+2+exts(rel8 × 2) → PC	
BNV:D label9	D	F9	1	----	-	if (V==0) then PC+2+exts(rel8 × 2) → PC	
BLT:D label9	D	FA	1	----	-	if (V ^ N==1) then PC+2+exts(rel8 × 2) → PC	
BGE:D label9	D	FB	1	----	-	if (V ^ N==0) then PC+2+exts(rel8 × 2) → PC	
BLE:D label9	D	FC	1	----	-	if ({V ^ N} Z==1) then PC+2+exts(rel8 × 2) → PC	
BGT:D label9	D	FD	1	----	-	if ({V ^ N} Z==0) then PC+2+exts(rel8 × 2) → PC	
BLS:D label9	D	FE	1	----	-	if (C or Z==1) then PC+2+exts(rel8 × 2) → PC	
BHI:D label9	D	FF	1	----	-	if (C or Z==0) then PC+2+exts(rel8 × 2) → PC	

- Delayed Branching Instructions are branched after always executing the following Instruction (the Delay Slot).
- The field "rel8" in TYPE-D instruction format and the field "rel11" in TYPE-D format have the following relation to the values label9, label12 in assembly notation.

$$\text{rel8} = (\text{label9} - \text{PC} - 2) / 2$$

$$\text{rel11} = (\text{label12} - \text{PC} - 2) / 2$$

Table A.2-13 Direct Addressing Instructions (14 Instructions)

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	-	$(dir8 \times 4) \rightarrow R13$	Word
DMOV R13, @dir10	D	18	a	----	-	$R13 \rightarrow (dir8 \times 4)$	
DMOV @dir10, @R13+	D	0C	2a	----	-	$(dir8 \times 4) \rightarrow (R13),$ $R13+4 \rightarrow (R13)$	
DMOV @R13+, @dir10	D	1C	2a	----	-	$(R13) \rightarrow (dir8 \times 4),$ $R13+4 \rightarrow (R13)$	
DMOV @dir10, @-R15	D	0B	2a	----	-	$R15-4 \rightarrow (R15),$ $(dir8 \times 4) \rightarrow (R15)$	
DMOV @R15+, @dir10	D	1B	2a	----	-	$(R15) \rightarrow (dir8 \times 4),$ $R15+4 \rightarrow (R15)$	
DMOVH @dir9, R13	D	09	b	----	-	$(dir8 \times 2) \rightarrow R13$	Half-Word
DMOVH R13, @dir9	D	19	a	----	-	$R13 \rightarrow (dir8 \times 2)$	
DMOVH @dir9, @R13+	D	0D	2a	----	-	$(dir8 \times 2) \rightarrow (R13),$ $R13+2 \rightarrow (R13)$	
DMOVH @R13+, @dir9	D	1D	2a	----	-	$(R13) \rightarrow (dir8 \times 2),$ $R13+2 \rightarrow (R13)$	
DMOVB @dir8, R13	D	0A	b	----	-	$(dir8) \rightarrow R13$	Byte
DMOVB R13, @dir8	D	1A	a	----	-	$R13 \rightarrow (dir8)$	
DMOVB @dir8, @R13+	D	0E	2a	----	-	$(dir8) \rightarrow (R13),$ $R13+2 \rightarrow (R13)$	
DMOVB @R13+, @dir8	D	1E	2a	----	-	$(R13) \rightarrow (dir8),$ $R13+2 \rightarrow (R13)$	

- The field "dir8" in TYPE-D Instruction format has the following relation to the values of dir8, dir9, dir10 in assembly notation.

dir8 = dir8

dir8 = dir9 >> 1

dir8 = dir10 >> 2

MB91460 Series**Table A.2-14 Other Instructions (16 Instructions)**

Mnemonic	Format	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
NOP	E'	9F-A	1	----	-	No change	
ANDCCR #u8	D	83	c	CCCC	-	CCR & u8 → CCR	
ORCCR #u8	D	93	c	CCCC	-	CCR u8 → CCR	
STILM #u8	D	87	1	----	-	u8 → ILM	Sets ILM immediate value
ADDSP #s10	D	A3	1	----	-	$R15 + s8 \times 4 \rightarrow R15$	
EXTSB Ri	E	97-8	1	----	-	$\text{exts}(Ri[7:0]) \rightarrow Ri$	Sign extension 8 → 32
EXTUB Ri	E	97-9	1	----	-	$\text{extu}(Ri[7:0]) \rightarrow Ri$	Zero extension 8 → 32
EXTSH Ri	E	97-A	1	----	-	$\text{exts}(Ri[15:0]) \rightarrow Ri$	Sign extension 16 → 32
EXTUH Ri	E	97-B	1	----	-	$\text{extu}(Ri[15:0]) \rightarrow Ri$	Zero extension 16 → 32
LDM0 (reglist)	D	8C	*1	----	-	$(R15) \rightarrow \text{reglist}, R15+4 \rightarrow R15$	Load Multiple R0 to R7
LDM1 (reglist)	D	8D	*1	----	-	$(R15) \rightarrow \text{reglist}, R15+4 \rightarrow R15$	Load Multiple R8 to R15
STM0 (reglist)	D	8E	*2	----	-	$R15-4 \rightarrow R15, \text{reglist} \rightarrow (R15)$	Store multiple R0 to R7
STM1 (reglist)	D	8F	*2	----	-	$R15-4 \rightarrow R15, \text{reglist} \rightarrow (R15)$	Store multiple R8 to R15
ENTER #u10	D	0F	1+a	----	-	$R14 \rightarrow (R15-4), R15-4 \rightarrow R14, R15-\text{extu}(u8 \times 4) \rightarrow R15$	Function entry processing
LEAVE	E'	9F-9	b	----	-	$R14+4 \rightarrow R15, (R15-4) \rightarrow R14$	Function exit processing
XCHB @Rj, Ri	A	8A	2a	----	m	$Ri \rightarrow \text{TEMP}, \text{extu}((Rj)) \rightarrow Ri, \text{TEMP} \rightarrow (Rj)$	Byte data for semaphore processing

*1: The number of execution cycles for LDM0 (reglist) and LDM1 (reglist) is $a \times (n-1) + 1$ cycles when "n" is the number of registers designated.

*2: The number of execution cycles for STM0 (reglist) and STM1 (reglist) is $a \times n + 1$ when "n" is the number of registers designated.

- In the ADDSP Instruction, the field s8 in TYPE-D Instruction Format has the following relation to the value of s10 in assembly notation.

$$s8 = s10 \gg 2$$

- In the ENTER Instruction, the field u8 in TYPE-D Instruction Format has the following relation to the value of u10 in assembly notation.

$$u8 = u10 \gg 2$$

Table A-15 Resource Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	Operation	Remarks
LDRES @Ri+, #u4	C	BC	a	----	(Ri) → u4 resource Ri+=4	u4: Channel number
STRES #u4, @Ri+	C	BD	a	----	u4 resource → (Ri) Ri+=4	u4: Channel number

Note:

This series cannot use these instructions because it has no resource with the channel number used for the resource instructions.

Table A-16 Coprocessor Control Instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	Operation	Remarks
COPOP #u4, #u8, CRj, CRi	E	9F-C	2+a	----	Operation instruction	No error trap
COPLD #u4, #u8, Rj, CRi	E	9F-D	1+2a	----	Rj → CRi	
COPST #u4, #u8, CRj, Ri	E	9F-E	1+2a	----	CRj → Ri	
COPSV #u4, #u8, CRj, Ri	E	9F-F	1+2a	----	CRj → Ri	

Note:

Since this series has no coprocessor, these instructions cannot be used.

MB91460 Series

A.3 Instruction Maps

Instruction maps are as follows.

Table A.3-1 Instruction Map

Table A.3-1 illustrates in tabular form 8-bit operation codes (OP) for each instruction. Instructions where operation code (OP) is less than 8 bits, they have been converted into 8 bit by packing them to MSB side.

Higher 4 bits

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	LD @ (R13,Ri), Ri	ST Ri, @ (R13,Ri)							BANDL #u4, @Ri	BORL #u4, @Ri	ADDN# #u4, Ri	LSR #u4, Ri			BRA:D label9	BRA:D label9
1	LDUH @ (R13,Ri), Ri	STH Ri, @ (R13,Ri)							BANDH #u4, @Ri	BORH #u4, @Ri	ADDN2 #u4, Ri	LSR2 #u4, Ri			BNO:D label9	BNO:D label9
2	LDUB @ (R13,Ri), Ri	STB Ri, @ (R13,Ri)							AND Ri, Ri	OR Ri, Ri	ADDNRi, Ri	LSR Ri, Ri			BEQ:D label9	BEQ:D label9
3	LD @ (R15, udisp6), Ri	ST Ri, @ (R15,ud6)							ANDCCR #u8	ORCCR #u8	ADDSP #s10	MOV Ri, Rs			BNE:D label9	BNE:D label9
4	LD @Ri,Ri	ST Ri, @Ri							AND Ri, @Ri	OR Ri, @Ri	ADD #u4, Ri	LSL #u4, Ri			BC:D label9	BC:D label9
5	LDUH @Ri,Ri	STH Ri, @Ri							ANDH Ri, @Ri	ORH Ri, @Ri	ADD2 #u4, Ri	LSL2 #u4, Ri			BNC:D label9	BNC:D label9
6	LDUB @Ri,Ri	STB Ri, @Ri							ANDB Ri, @Ri	ORB Ri, @Ri	ADD Ri, Ri	LSL Ri, Ri			BN:D label9	BN:D label9
7	E format	E format	LD @ (R14, disp10), Ri	ST Ri, @ (R14, disp10)	LDUH @ (R14, disp9), Ri	STH Ri, @ (R14, disp9)	LDUB @ (R14, disp8), Ri	STB Ri, @ (R14, disp8)	STLM #u8	E format	ADDCRi, Ri	MOV Rs, Ri			BP:D label9	BP:D label9
8	DMOV @d10, R13	DMOV R13, @d10							BTSTL #u4, @Ri	BEORL #u4, @Ri	CMP #u4, Ri	ASR #u4, Ri			BV:D label9	BV:D label9
9	DMOVH @d9, R13	DMOVH R13, @d9							BTSTH #u4, @Ri	BEORH #u4, @Ri	CMP2 #u4, Ri	ASR2 #u4, Ri			BNV:D label9	BNV:D label9
A	DMOV @ d8, R13	DMOV R13, @d8							XCHB @Ri, Ri	EOR Ri, Ri	CMP Ri, Ri	ASR Ri, Ri			BLT:D label9	BLT:D label9
B	DMOV @d10, @R15	DMOV @R15+, @d10							MOV Ri, Ri	LD:20 #20, Ri	MULU Ri, Ri	MULUH Ri, Ri			BGE:D label9	BGE:D label9
C	DMOV @d10, @R13+	DMOV @R13+, @d10							LDM0 (reglist)	EORRi, @Ri	SUB Ri, Ri	LDRES @Ri+, #u4			BLE:D label9	BLE:D label9
D	DMOVH @d9, @R13+	DMOVH @R13+, @d9							LDM1 (reglist)	EORH Ri, @Ri	SUBCRi, Ri	STRES @Ri+, #u4			BGT:D label9	BGT:D label9
E	DMOV @d8, @R13+	DMOV @R13+, @d8							STM0 (reglist)	EOR Ri, @Ri	SUBN Ri, Ri				BLS:D label9	BLS:D label9
F	ENTER #u10	INT #u8							STM1 (reglist)	E format	MUL Ri, Ri	MULH Ri, Ri			BHI:D label9	BHI:D label9

Lower 4 bits

A.4 Instruction Maps of Instruction Format TYPE-E

Instruction Maps of TYPE-E and TYPE-E' instruction formats are illustrated.

Table A.4-1 illustrates in tabular form 8-bit operation codes (OP) and 4-bit sub-operation codes (SUB-OP) for each instruction.

Table A.4-1 Instruction Map of Instruction Format TYPE-E

		Higher 8 bits			
		07	17	97	9F
Lower 4 bits	0	LD @R15+,Ri	ST Ri,@-R15	JMP @Ri	JMP:D @Ri
	1	MOV Ri,PS	MOV PS,Ri	CALL @Ri	CALL:D @Ri
	2	-	-	RET	RET:D
	3	-	-	RETI	INTE
	4	-	-	DIV0S Ri	-
	5	-	-	DIV0U Ri	-
	6	-	-	DIV1 Ri	DIV3
	7	-	-	DIV2 Ri	DIV4S
	8	LD @R15+,Rs	ST Rs,@-R15	EXTSB Ri	LDI:32 #i32,Ri
	9	LD @R15+,PS	ST PS,@-R15	EXTUB Ri	LEAVE
	A	-	-	EXTSH Ri	NOP
	B	-	-	EXTUH Ri	COPOP #u4,#CC,CRj,CRI
	C	-	-	-	COPLD #u4,#CC,Rj,CRI
	D	-	-	-	COPST #u4,#CC,CRj,Ri
	E	-	-	-	COPSV #u4,#CC,CRj,Ri

-: Undefined

MB91460 Series

MB91460 Series**Appendix B List of Figures**

Figure 1.1-1	MB91460 series roadmap	1
Figure 1.2-1	Opposite phase external clock driving	3
Figure 2.4-1	Block Diagram MB91FV460B	30
Figure 3.1-1	Memory Map	35
Figure 3.4-1	External Dimension of BGA-660P-M02	121
Figure 3.4-2	External Dimension of BGA-896P-M02	122
Figure 3.5-1	Pin Assignment Diagram of MB91V460A	123
Figure 3.5-2	Pin Assignment Diagram of MB91FV460B, top view	124
Figure 3.5-3	Pin Assignment Diagram of MB91FV460B, bottom view	125
Figure 4.1-1	Connection Diagram of Internal Architecture (Core Group)	201
Figure 4.3-1	Instruction Pipeline	203
Figure 4.7-1	Bit Structure of Bit Ordering	205
Figure 4.7-2	Bit Structure of Byte Ordering	205
Figure 4.8-1	[Example] LD @(R13, R2), R0	206
Figure 4.9-1	Address Map	207
Figure 5.1-1	General-purpose Registers	209
Figure 5.2-1	Dedicated Registers List	209
Figure 5.2-2	Bit Structure of Program Counter (PC)	210
Figure 5.2-3	Bit Structure of Program Status (PS)	210
Figure 5.2-4	Structure of Condition Code Register (CCR)	210
Figure 5.2-5	Structure of System Condition Code Register (SCR)	212
Figure 5.2-6	Register Structure of Interrupt Level Mask Register (ILM)	212
Figure 5.2-7	Bit Structure of Table-base Register (TBR)	214
Figure 5.2-8	Bit Structure of Return Pointer (RP)	214
Figure 5.2-9	Bit Structure of System Stack Pointer (SSP)	214
Figure 5.2-10	Bit Structure of User Stack Pointer (USP)	215
Figure 5.2-11	Interrupt Stack	215
Figure 5.2-12	Bit Structure of Multiply & Divide Register (MDH/MDL)	216
Figure 6.8-1	Multiple EITs Process	221
Figure 7.3-1	Example (Branch instruction with delay slot)	227
Figure 7.4-1	Example	228
Figure 7.4-2	Example	228
Figure 7.4-3	Example	228
Figure 7.4-4	Example	228
Figure 7.7-1	Example (Branch Instruction without Delay Slot)	230
Figure 8.1-1	Overview Diagram of Device State Transition	231
Figure 8.3-1	State Transition of MB91460 Series	232
Figure 9.1-1	Flow of Reset Operation	237
Figure 9.2-1	External reset / initialization chain	238
Figure 9.3-1	Configuration Diagram of Reset operation	239
Figure 9.3-2	Register List	239
Figure 10.3-1	Configuration Diagram of Standby operation	254
Figure 10.3-2	Register List	254
Figure 11.8-1	Timing of a FLASH access cycle	273
Figure 12.2-1	Instruction Cache Structure	275
Figure 12.3-1	Instruction Cache Tag Structure	276
Figure 12.5-1	Memory Allocation for Cache Capacities	278
Figure 12.5-2	I-RAM allocation when ICHCR.RAM=0	279
Figure 12.5-3	RAM allocation when ICHCR.RAM=1	280
Figure 12.5-4	RAG-RAM mirror when ICHCR.RAM=1	281
Figure 12.5-5	Cache RAM / I-RAM mirror	282
Figure 12.5-6	RAG-RAM and Cache RAM configuration	283

Figure 12.6-1	I-Cacheable Area.....	285
Figure 13.1-1	Block Diagram of Clock Distribution	289
Figure 13.3-1	Configuration Diagram of the clock controller.....	291
Figure 13.3-2	Configuration Diagram of the Main- and Sub-Oscillators	292
Figure 13.4-1	Clock Control Register List	292
Figure 13.5-1	Clock Setup Sequence (Example).....	304
Figure 13.5-2	Halting and Restarting the Main Clock Oscillation During Sub Clock Mode (Example)	305
Figure 14.2-1	Block diagram of Main PLL interface	313
Figure 14.3-1	E-Ray PLL Interface	315
Figure 14.3-2	USB PLL Interface	316
Figure 15.1-1	Combination of CAN Clock prescaler and CAN interface.....	329
Figure 16.1-1	Block diagram of the clock supervisor	333
Figure 16.2-1	Configuration Clock Supervisor Control Register (CSVCR)	335
Figure 16.3-1	Block Diagram of Clock Supervisor.....	337
Figure 16.4-1	Timing Diagram: Initial settings, Main clock missing during power-on reset	339
Figure 16.4-2	Timing Diagram: Initial settings, Main clock missing during 'oscillation stabilization wait time'	340
Figure 16.4-3	Timing Diagram: Initial settings, Main clock missing after 'oscillation stabilization wait time' ..	341
Figure 16.4-4	Timing Diagram: Initial settings, Sub clock missing before timeout.....	342
Figure 16.4-5	Timing Diagram: Initial settings, Sub clock missing after timeout.....	343
Figure 16.4-6	Timing Diagram: Disabling the RC-oscillator and the clock supervisors	344
Figure 16.4-7	Timing Diagram: Re-enabling the RC-oscillator and the clock supervisors.....	345
Figure 16.4-8	Timing Diagram: Sub clock missing in Main clock mode, SRST=0	347
Figure 16.4-9	Timing Diagram: Sub clock missing in Main clock mode, SRST=1	348
Figure 16.4-10	Timing Diagram: Waking up from Sub clock mode	349
Figure 16.4-11	Timing Diagram: Waking up from STOP state	350
Figure 16.4-12	Timing Diagram: Sub clock mode with single clock device	353
Figure 17.1-1	Block diagram of the Clock modulator	355
Figure 17.1-2	Frequency spectrum of the modulated clock.....	355
Figure 17.2-1	Clock modulator registers.....	356
Figure 17.2-2	Configuration of the clock modulator control register (CMCR)	357
Figure 17.2-3	Modulation parameter register.....	360
Figure 17.2-4	Skew of modulated clock vs. unmodulated clock	361
Figure 18.1-1	Timebase Counter (overview diagram)	363
Figure 18.1-2	Timebase counter when used to generate the oscillation stabilization wait	363
Figure 18.3-1	Configuration Diagram of the timebase counter for oscillation stabilization wait time	365
Figure 18.3-2	Register List.....	365
Figure 18.5-1	Using the Width of the Pin Input to Provide the Oscillation Stabilization Wait Time.....	369
Figure 18.5-2	Watchdog Reset when Main Clock Halted (Sub-RUN)	370
Figure 18.5-3	Recovering from STOP state to Main PLL Operation via an Interrupt.....	371
Figure 18.5-4	Stabilization Wait for each state transition (necessary or not).....	374
Figure 19.1-1	Timebase Counter (overview diagram)	379
Figure 19.1-2	Timebase Counter used to generate the Timebase Timer Interrupt.....	379
Figure 19.3-1	Configuration Diagram of the Timebase Timer.....	380
Figure 19.3-2	List of Registers.....	380
Figure 19.5-1	Example for Timebase Timer Interrupt (Main PLL Lock Wait).....	383
Figure 20.1-1	Timebase Counter (overview diagram)	389
Figure 20.1-2	Timebase Counter used to generate the Watchdog Reset	389
Figure 20.3-1	Configuration Diagram of the Watchdog Timer	390
Figure 20.3-2	List of Registers.....	390
Figure 20.5-1	Example of Watchdog operation (Detecting Runaway).....	394
Figure 20.8-1	One bit counter reset	398
Figure 21.1-1	Block Diagram of hardware watchdog timer.....	400
Figure 21.1-2	External reset / initialization chain	400
Figure 22.1-1	Counter used to generate Main clock oscillation stabilization wait interrupt.....	405

MB91460 Series

Figure 22.3-1	Configuration Diagram of the Main clock oscillation stability wait time selection	406
Figure 22.3-2	List of Registers	406
Figure 22.5-1	Example operation of Main Clock Oscillation Stability Wait	409
Figure 22.5-2	Example of generating an interval interrupt	410
Figure 22.6-1	Settings Required for Using the Main Clock Oscillation Stability Wait Timer	411
Figure 22.6-2	Settings Required for Enabling the Main Clock Oscillation Stability Wait Timer Interrupt	411
Figure 22.6-3	Settings Required for Stopping the Main Clock Oscillation Stability Wait Timer	411
Figure 23.1-1	Counter used to generate Sub clock oscillation stabilization wait interrupt	417
Figure 23.3-1	Configuration Diagram of the Sub clock oscillation stabilization timer	418
Figure 23.3-2	List of Registers	418
Figure 23.5-1	Example of generating a Sub clock Oscillation Stability Wait Interrupt	420
Figure 23.5-2	Example of generating an Interval Interrupt (Clock Interrupt)	421
Figure 23.5-3	Returning from the STOP state due to Interval Operation (Clock Interrupt)	422
Figure 24.1-1	Block diagram of Interrupt Control	429
Figure 24.3-1	Configuration Diagram of Interrupt Control	430
Figure 24.3-2	Configuration Diagram of Interrupt Control (CPU side)	430
Figure 24.5-1	Priority determination operation of interrupt control	435
Figure 24.5-2	Hardware configuration for Hold request cancel request	437
Figure 24.5-3	Interrupt level $HRCL > a$	437
Figure 24.5-4	Interrupt level $HRCL > a > b$	437
Figure 25.1-1	Block diagram of external interrupt	441
Figure 25.3-1	Configuration Diagram for external interrupts 0-7	442
Figure 25.3-2	Configuration Diagram for external interrupts 8-15	443
Figure 25.3-3	Configuration Diagram for external interrupts 16-23	444
Figure 25.3-4	Configuration Diagram for external interrupts 24-31	445
Figure 25.3-5	Register List External Interrupts 0-7	446
Figure 25.3-6	Register List External Interrupts 8-15	447
Figure 25.3-7	Register List External Interrupts 16-23	448
Figure 25.3-8	Register List External Interrupts 24-31	449
Figure 25.5-1	External Interrupt: Level and Edge Detection	457
Figure 25.5-2	External Interrupt: Wakeup from STOP State	458
Figure 25.5-3	Timing of consecutive NMI events	459
Figure 26.1-1	Block Diagram of the DMA Controller (DMAC)	466
Figure 26.2-1	DMA Controller (DMAC) Registers	467
Figure 26.2-2	Bit Configuration of Control/Status Registers A (DMACA0 to 4)	468
Figure 26.2-3	Bit Configuration of Control/Status Registers B (DMACB0 to 4)	478
Figure 26.2-4	Bit Configuration of the Transfer Source/Transfer Destination Address Setting Registers	485
Figure 26.2-5	Bit Configuration of the DMAC All-Channel Control Register (DMACR)	486
Figure 26.3-1	Example of burst transfer for a start on an external pin rising edge	493
Figure 26.3-2	Example of demand transfer for a start with the external pin at H level	493
Figure 26.3-4	Example of block transfer for a start for an external pin on a rising edge	495
Figure 26.3-5	Timing Example in Fixed Mode	503
Figure 26.3-6	Timing Example in Rotation Mode	503
Figure 26.3-7	Negate timing example of the DREQ pin input for 2-cycle external --> internal	505
Figure 26.3-8	Negate timing example of the DREQ pin input for fly-by (write) transfer	505
Figure 26.4-1	Operation Flowchart for Block Transfer	508
Figure 26.4-2	Operation Flowchart for Burst Transfer	509
Figure 26.4-3	Operation Flowchart for Demand Transfer	510
Figure 26.5-1	Examples of 2-Cycle Transfer (Continued on next page)	511
Figure 26.5-2	Examples of Fly-By Transfer	513
Figure 26.6-1	Timing Chart for Transfer Other Than the Demand Transfer	515
Figure 26.6-2	Timing Chart for Demand Transfer	515
Figure 27.1-1	Block diagram of delayed Interrupt	517
Figure 27.3-1	Configuration diagram of delayed interrupt	517

Figure 27.3-2	List of Registers	517
Figure 27.5-1	Delayed interrupt service	518
Figure 28.1-1	Block diagram of the Bit search module	521
Figure 28.3-1	Configuration Diagram of the bit search module	522
Figure 28.3-2	List of Registers	522
Figure 28.5-1	Example of zero detection	525
Figure 28.5-2	Example of one detection	525
Figure 28.5-3	Example of changing position detection	526
Figure 29.1-1	EDSU block diagram	532
Figure 30.4-1	Block diagram of an I/O port	595
Figure 31.1-1	Block diagram of External bus interface	655
Figure 31.1-2	List of External Bus Interface Registers	658
Figure 31.2-1	Configuration of the Area Select Registers (ASR0-7)	660
Figure 31.2-2	Configuration of Area Configuration Registers 0-7 (ACR0-7) (Continued on next page)	661
Figure 31.2-3	Configuration of the Area Wait Registers (AWR0-7) (Continued on next page)	667
Figure 31.2-4	Bit configuration of the MCRA register (for SDRAM/FCRAM auto - precharge OFF mode) ...	675
Figure 31.2-5	Structure of the Memory Setting Register (MCRB for FCRAM auto - precharge ON mode)...	676
Figure 31.2-6	Configuration of the I/O Wait Registers for DMAC (IOWR0-3)	677
Figure 31.2-7	Configuration of the Chip Select Enable Register (CSER)	679
Figure 31.2-8	Configuration of the Cache Enable Register (CHER)	680
Figure 31.2-9	Configuration of the Pin/Timing Control Register (TCR)	681
Figure 31.2-10	Structure of the Refresh Control Register (RCR)	683
Figure 31.3-1	Example of Setting the Chip Select Area	686
Figure 31.4-1	Data Bus Width and Control Signal on the Ordinary Bus Interface	687
Figure 31.4-2	Data Bus Width and Control Signal in the Time Division I/O Interface	687
Figure 31.4-3	Data bus width of the SDRAM (FCRAM) interface and its control signals	688
Figure 31.4-4	Relationship between Internal Register and External Data Bus for Word Access	688
Figure 31.4-5	Relationship between the Internal Register and External Data Bus for Halfword Access	689
Figure 31.4-6	Relationship between Internal Register and External Data Bus for Byte Access	689
Figure 31.4-7	Relationship between Internal Register and External Bus Having 32-Bit Bus Width	689
Figure 31.4-8	Relationship between Internal Register and External Bus Having 16-Bit Bus Width	690
Figure 31.4-9	Relationship between Internal Register and External Bus having 8-Bit bus Width	690
Figure 31.4-10	External bus Access for 32-Bit Bus Width	691
Figure 31.4-11	External bus Access for 16-Bit Bus Width	691
Figure 31.4-12	External Bus Access for 8-Bit Bus Width	692
Figure 31.4-13	Example of Connecting the MB91460 Series to External Devices	693
Figure 31.4-14	Relationship between the Internal Register and External Data Bus for Word Access	694
Figure 31.4-15	Relationship between Internal Register and External Data Bus for Halfword Access	694
Figure 31.4-16	Relationship between Internal Register and External Data Bus for Byte Access	695
Figure 31.4-17	Relationship between Internal Register and External Bus Data for 32-bit Bus Width	695
Figure 31.4-18	Relationship between Internal Register and External Bus Data for 16-bit Bus Width	695
Figure 31.4-19	Relationship between the Internal Register and External Data Bus in the 8-bit Bus Width	696
Figure 31.4-20	Example of Connecting the MB91460 Series to External Devices (32-Bit Bus Width)	696
Figure 31.4-21	Example of Connecting the MB91460 Series to External Devices (16-Bit Bus Width)	697
Figure 31.4-22	Example of Connecting the MB91460 Series to External Devices (8-Bit Bus Width)	697
Figure 31.4-23	Comparison of Big Endian and Little Endian External Access (Word access)	698
Figure 31.4-24	Comparison of Big Endian and Little Endian External Access (Halfword access)	699
Figure 31.4-25	Comparison of Big Endian and Little Endian External Access (Byte access)	701
Figure 31.5-1	Basic Timing (For Successive Accesses)	705
Figure 31.5-2	Timing Chart for the WRn + Byte Control Type	706
Figure 31.5-3	Timing Chart for Read -> Write	707
Figure 31.5-4	Timing Chart for the Write -> Write Operation	708
Figure 31.5-5	Timing Chart for the Auto-Wait Cycle	709
Figure 31.5-6	Timing Chart for the External Wait Cycle	710

MB91460 Series

Figure 31.5-7	Timing Chart for Synchronous Write Enable Output	711
Figure 31.5-8	Operation Timing Chart for the CSX Delay Setting	712
Figure 31.5-9	Timing Chart for the CSXn -> RDX/WRXn/WEX Setup and RDX/WRXn/WEX -> CSXn Hold Settings	713
Figure 31.5-10	Timing Chart for DMA Fly-By Transfer (I/O -> Memory).....	714
Figure 31.5-11	Timing Chart for DMA Fly-By Transfer (Memory -> I/O).....	715
Figure 31.6-1	Timing Chart for Burst Access	716
Figure 31.7-1	Timing Chart for the Address/Data Multiplex Interface (without External Wait).....	718
Figure 31.7-2	Timing Chart for the Address/Data Multiplex Interface (with External Wait).....	719
Figure 31.7-3	Timing Chart for the Address/Data Multiplex Interface (CSXn -> RD/WRn Setup)	720
Figure 31.9-1	Burst Read/Write Timing Chart.....	724
Figure 31.9-2	Single Read/Write Timing Chart	725
Figure 31.9-3	Single Read Timing Chart	725
Figure 31.9-4	Single Read/Write Timing Chart	726
Figure 31.9-5	Auto - refresh Timing Chart	726
Figure 31.9-6	Examples of combinations of access addresses and Row/BANK/Column addresses	729
Figure 31.9-7	Using 64 - Mbit SDRAM	731
Figure 31.9-8	Using 64 - Mbit SDRAM	732
Figure 31.9-9	Using 64 - Mbit	732
Figure 31.10-1	Timing Chart for DMA Fly-By Transfer (I/O -> Memory).....	734
Figure 31.10-2	Timing chart for DMA Fly-By Transfer (Memory -> I/O).....	735
Figure 31.10-3	Timing Chart for DMA Fly - by Transfer (I/O to SDRAM/FCRAM).....	736
Figure 31.10-4	Timing Chart for DMA Fly - by Transfer (SDRAM/FCRAM to I/O) with Page Hits (Shortest) ..	738
Figure 31.10-5	Timing Chart for DMA Fly - by Transfer (SDRAM/FCRAM to I/O) with Page Misses.....	739
Figure 31.10-6	Sample Circuit Solving a Fly-by Penalty Using External Wait Cycles Based on CAS.....	740
Figure 31.10-7	Timing Chart for Fly - by Penalty Solution Using External Wait Cycles Based CAS	741
Figure 31.10-8	Timing Chart for 2-cycle Transfer (Internal RAM -> External I/O, RAM)	742
Figure 31.10-9	Timing Chart for 2-Cycle Transfer (External -> I/O).....	743
Figure 31.10-10	Timing Chart for 2-Cycle Transfer (I/O -> External).....	744
Figure 31.10-11	Timing Chart for Two - cycle Transfer (I/O to SDRAM/FCRAM)	745
Figure 31.10-12	Timing Chart for Two - cycle Transfer (SDRAM/FCRAM to I/O)	746
Figure 31.11-1	Timing Chart for Releasing the Bus Right	747
Figure 31.11-2	Timing Chart for Acquiring the Bus Right	747
Figure 32.2-1	USART Block Diagram	755
Figure 32.3-1	Block Diagram of USART pins	758
Figure 32.4-1	Serial Control Register 04 (SCR04)	760
Figure 32.4-2	Configuration of the Serial Mode register 04 (SMR04).....	762
Figure 32.4-3	Configuration of the Serial Status register 04 (SSR04)	764
Figure 32.4-4	Transmission and Reception Data registers 04 (RDR04 / TDR04)	766
Figure 32.4-5	Configuration of the Extended Status/Control Register (ESCR04)	767
Figure 32.4-6	Configuration of the Extended Communication Control Register (ECCR04)	769
Figure 32.4-7	Baudrate Reload Counter Register 0 and 1 (BGR04 / 14)	771
Figure 32.4-8	Configuration of FIFO control register	772
Figure 32.4-9	Configuration of FIFO status register	774
Figure 32.5-1	Bus idle interrupt generation.....	778
Figure 32.5-2	Reception operation and flag set timing	779
Figure 32.5-3	ORE set timing	779
Figure 32.5-4	Transmission operation and flag set timing	780
Figure 32.6-1	Baud rate selection circuit (reload counter)	782
Figure 32.6-2	Counting example of the reload counters.....	785
Figure 32.6-3	Reload Counter Restart example	786
Figure 32.7-1	Transfer data format (operation modes 0 and 1).....	788
Figure 32.7-2	Transfer data format (operation mode 2).....	790
Figure 32.7-3	SPI Transfer data format (operation mode 2).....	790

Figure 32.7-4	Transfer data format with clock inversion	791
Figure 32.7-5	Continuous clock output in mode 2	791
Figure 32.7-6	USART behavior as slave in LIN mode	793
Figure 32.7-7	LIN bus timing and USART signals	794
Figure 32.7-8	Settings for USART operation mode 0 and 2	795
Figure 32.7-9	Connection example of USART mode 2 bidirectional communication	795
Figure 32.7-10	Settings for USART operation mode 1	796
Figure 32.7-11	Connection example of USART master-slave communication	796
Figure 32.7-12	Master-slave communication flowchart	798
Figure 32.7-13	Settings for USART	799
Figure 32.7-14	Connection example of a small LIN-Bus system	799
Figure 32.7-15	USART LIN master flow chart	800
Figure 32.7-16	USART LIN slave flow chart (part1)	801
Figure 32.7-17	USART LIN slave flow chart (part 2)	802
Figure 32.8-1	Baud Rate Detection Using the Input Capture Units	804
Figure 32.8-2	Timing of the CRE bit	805
Figure 32.8-3	Data Stream Synchronization	805
Figure 32.8-4	USART Dominant Bus Behaviour.....	806
Figure 33.1-1	Block diagram of I2C Controller.....	808
Figure 33.2-1	Bus Control Register (IBCR0)	809
Figure 33.2-2	Bus Status Register (IBSR0)	809
Figure 33.2-3	Ten Bit slave Address register (ITBA0)	809
Figure 33.2-4	Ten bit slave address MasK register (ITMK0)	809
Figure 33.2-5	Seven bit slave address MasK register (ISMK0)	810
Figure 33.2-6	Seven Bit slave Address register (ISBA0)	810
Figure 33.2-7	Data Register (IDAR0).....	810
Figure 33.2-8	Clock control register (ICCR0).....	810
Figure 33.2-9	Bus Control Register (IBCR0)	811
Figure 33.2-10	Bus Status Register (IBSR0)	815
Figure 33.2-11	Ten Bit Slave Address Register (ITBA0)	818
Figure 33.2-12	Ten Bit Address Mask Register (ITMK0)	819
Figure 33.2-13	Seven Bit Slave Address Register (ISBA0).....	820
Figure 33.2-14	Seven Bit Slave Address Mask Register (ISMK0).....	821
Figure 33.2-15	Data Register (IDAR0).....	822
Figure 33.2-16	Clock Control Register (ICCR0)	823
Figure 33.2-17	SCL Waveforms	824
Figure 33.4-1	Example Of Slave Addressing And Sending Data.....	827
Figure 33.4-2	Example Of Receiving Data	828
Figure 33.4-3	Example Of An Interrupt Handler	829
Figure 34.1-1	Block diagram of the CAN Controller.....	831
Figure 34.2-1	CAN Prescaler Register Summary	837
Figure 34.2-2	CAN Control Register (CTRLR).....	837
Figure 34.2-3	Status Register (STATR).....	839
Figure 34.2-4	Error Counter (ERRCNT)	841
Figure 34.2-5	Bit Timing Register (BTR).....	842
Figure 34.2-6	Test Register (TESTR)	843
Figure 34.2-7	BRP Extension Register (BRPER)	845
Figure 34.2-8	IF1 and IF2 Message Interface Register Sets	846
Figure 34.2-9	IFx Command Request Registers (IFxCREQ).....	846
Figure 34.2-10	IFx Command Mask Register (IFxCMSK)	847
Figure 34.2-11	IFx Mask 2 Register (IFxMSK2)	850
Figure 34.2-12	IFx Mask 1 Register (IFxMSK1)	850
Figure 34.2-13	IFx Arbitration 2 Register (IFxARB2)	850
Figure 34.2-14	IFx Arbitration 1 Register (IFxARB1)	851

MB91460 Series

Figure 34.2-15	IFx Message Control Register (IFxMCTR)	851
Figure 34.2-16	Structure of a Message Object in the Message Memory	852
Figure 34.2-17	Interrupt Register (INTR)	855
Figure 34.2-18	Transmission Request Register 2 (TREQR2)	856
Figure 34.2-19	Transmission Request Register 1 (TREQR1)	857
Figure 34.2-20	New Data Register 2 (NEWDT2)	858
Figure 34.2-21	New Data Register 1 (NEWDT1)	858
Figure 34.2-22	Interrupt Pending Register 2 (INTPND2)	859
Figure 34.2-23	Interrupt Pending Register 1 (INTPND1)	859
Figure 34.2-24	Message Valid Register 2 (MSGVAL2)	860
Figure 34.2-25	Message Valid Register 1 (MSGVAL1)	860
Figure 34.3-1	CAN Core in Silent Mode	863
Figure 34.3-2	CAN Core in Loop Back Mode	864
Figure 34.3-3	CAN Core in Loop Back combined with Silent Mode	864
Figure 34.4-1	Data Transfer between IFx Registers and Message RAM	867
Figure 34.4-2	Initialization of a Transmit Object	869
Figure 34.4-3	Initialization of a Receive Object	870
Figure 34.4-4	CPU Handling of a FIFO Buffer	872
Figure 34.4-5	Bit Timing.....	874
Figure 35.1-1	Block diagram of Free-Run Timer	875
Figure 35.3-1	Configuration Diagram of Free-Run Timer	876
Figure 35.3-2	Register List	877
Figure 35.6-1	Count Operation of the Free-Run Timer	882
Figure 35.6-2	Various Clear Operations of the Free-Run Timer	883
Figure 36.1-1	Block diagram of Input Capture (Example of channels 0/1)	889
Figure 36.2-1	Functionality of Input Capture.....	890
Figure 36.3-1	Configuration Diagram for Input Capture.....	890
Figure 36.3-2	Register List.....	890
Figure 36.5-1	Operation of Input Capture (Capture Timing, Interrupt Timing)	896
Figure 36.5-2	Input Capture Edge Specification and Operation	897
Figure 37.1-1	Structure of output compare (Example of channels 0/1)	903
Figure 37.2-1	Output waveform: Toggle output 4 channel.....	903
Figure 37.3-1	Configuration Diagram for Output compare.....	904
Figure 37.3-2	Register List.....	904
Figure 37.5-1	Output Compare Output (CMOD="0")	910
Figure 37.5-2	Output Compare Output (CMOD="1")	911
Figure 37.7-1	(Example) To output a period: A, phase difference 1/4 phase pulse	917
Figure 37.7-2	(Example) To output a period: A, duty 1/4 - 3/4 ("L") PWM,	917
Figure 37.8-1	Compare stop space	919
Figure 38.1-1	Block diagram of the Reload Timer	921
Figure 38.2-1	Operation: 2 types of operation are possible.....	921
Figure 38.3-1	Configuration Diagram of the Reload Timer (Internal Clock Count)	923
Figure 38.3-2	Configuration Diagram of the Reload Timer (External event count)	924
Figure 38.3-3	Register List.....	924
Figure 38.5-1	Operation of Reload Timer in reload mode (pulse with a 50% duty ratio is output)	929
Figure 38.5-2	Operation of Reload Timer in one-shot mode (one-shot pulse is output)	930
Figure 38.5-3	Operation in external event clock reload mode (outputs a pulse with a 50% duty ratio)	931
Figure 38.5-4	Operation in external event clock one-shot mode, (one-shot pulse is output).....	932
Figure 38.5-5	State Transition Diagram of the Reload Timer	933
Figure 39.1-1	Block diagram of PPG (Programmable Pulse Generator)	941
Figure 39.2-1	Output waveforms: The PPGs can generate the following six types of output signals.....	941
Figure 39.2-2	One-shot waveform (Rectangular wave)	941
Figure 39.3-1	Configuration Diagram of PPG 0-3.....	944
Figure 39.3-2	Register List.....	944

Figure 39.5-1	In PWM operation, variable duty cycle pulses are generated from the PPG pin	957
Figure 39.5-2	Restart available in PWM operation	960
Figure 39.5-3	Restart available in one-shot operation	960
Figure 39.7-1	PPG pin output can be set to all "L". (when OSEL="0")	964
Figure 39.7-2	PPG pin output can be set to all "H". (when OSEL="0")	965
Figure 40.1-1	16-bit Reload Counter 0 Register Configuration	975
Figure 40.1-2	16-bit Reload Counter 1 Register Configuration	976
Figure 40.1-3	Block Diagram of the 16-bit Pulse Frequency Modulator	977
Figure 40.2-1	Structure of the Control Status Register	978
Figure 40.2-2	Structure of the 16-bit Counter Register	981
Figure 40.2-3	Structure of the 16-bit Reload Register	981
Figure 40.3-1	Counter Activation and Operation Timing	982
Figure 40.3-2	Underflow Operation Timing	983
Figure 40.3-3	Counter State Transitions	984
Figure 41.1-1	Block diagram of an Up/Down Counter	987
Figure 41.3-1	Configuration Diagram of the Up/Down Counter 0 (8 bit mode)	988
Figure 41.3-2	Configuration Diagram of the Up/Down counter 1 (8 bit mode)	989
Figure 41.3-3	Configuration Diagram of the Up/Down Counter (16 bit mode)	990
Figure 41.3-4	Up/Down Counter 0 Register List (8-bit mode)	990
Figure 41.3-5	Up/Down Counter 1 Register List (8-bit mode)	991
Figure 41.3-6	Up/Down Counter 0/1 Register List (16-bit mode)	991
Figure 41.5-1	Up/Down Counter in Timer Mode CMS[1:0]="00"	1000
Figure 41.5-2	Up/Down Counter in Up/Down Count Mode CMS[1:0]="01"	1001
Figure 41.5-3	Up/Down Counter in Up/Down Count Mode CMS[1:0]="01"	1002
Figure 41.5-4	Up/Down Counter in Phase Difference Count Mode (Multiply by 2) CMS[1:0]="10"	1003
Figure 41.5-5	Up/Down Counter in Phase Difference Count Mode (Multiply by 4) CMS[1:0]="11"	1004
Figure 41.5-6	Clear Timing of Up/Down Counter (1)	1005
Figure 41.5-7	Clear Timing of Up/Down Counter (2)	1005
Figure 41.5-8	Clear Timing of Up/Down Counter (3)	1005
Figure 41.5-9	Clear Timing of Up/Down Counter (4)	1006
Figure 41.5-10	Reload Timing of Up/Down Counter	1006
Figure 41.5-11	Writing value to Up/Down Counter	1006
Figure 42.1-1	Block diagram of the sound generator	1017
Figure 42.2-1	Detailed Structure of the sound generator	1018
Figure 42.3-1	Sound Generator Register overview	1019
Figure 42.3-2	Sound Control Register (SGCR)	1020
Figure 42.3-3	Frequency Data Register (SGFR)	1021
Figure 42.3-4	Tone period of Sound Generator	1021
Figure 42.3-5	Amplitude Data Register (SAGR)	1022
Figure 42.3-6	Relationship between the register value and the PWM pulse	1022
Figure 42.3-7	Device auto disable register (SGDAD)	1023
Figure 42.3-8	Decrement Grade Register (SGDR)	1023
Figure 42.3-9	Tone Count Register (SGTR)	1024
Figure 43.1-1	Stepper Motor Controller with A/D Converter Function	1025
Figure 43.1-2	Stepper Motor Controller	1026
Figure 43.2-1	Stepper Motor Controller Register overview	1027
Figure 43.2-2	PWM Control Register	1028
Figure 43.2-3	PWM1 and PWM2 Compare Registers	1029
Figure 43.2-4	Relationship between the Compare Register Setting Value and PWM Pulse Width	1030
Figure 43.2-5	PWM1 and PWM2 Selection Registers	1031
Figure 43.2-6	load timing of PWM compare register value	1032
Figure 43.3-1	Setting of Stepper Motor Controller	1034
Figure 43.3-2	Examples of PWM1 and PWM2 Waveform Output	1035
Figure 44.1-1	A/D Converter sampling circuit	1038

MB91460 Series

Figure 44.2-1	A/D Converter block diagram	1039
Figure 44.3-1	A/D Converter ACH[5:0] Latch	1048
Figure 44.5-1	Total error of A/D converter	1055
Figure 44.5-2	ADC Nonlinearity and Differential Nonlinearity Error	1056
Figure 44.5-3	A/D Converter sampling replacement model	1057
Figure 44.6-1	A/D Converter Single-shot conversion (only 1 channel enabled)	1059
Figure 44.6-2	A/D converter Single-shot conversion (channels 0 to 3 enabled)	1060
Figure 44.6-3	A/D converter continous conversion (channels 0 to 3 enabled)	1061
Figure 44.6-4	A/D converter scan conversion with pause (channels 0 to 3 enabled)	1062
Figure 44.6-5	A/D Converter stop mode conversion	1064
Figure 44.7-1	A/D Converter interrupts	1065
Figure 44.8-1	A/D converter data protection	1066
Figure 44.12-1	A/D Converter Range Comparator block diagram	1077
Figure 44.14-1	ADC Range Comparator flag setting	1083
Figure 44.15-1	Range Comparator interrupt structure	1086
Figure 45.1-1	Block diagram of D/A converter	1089
Figure 45.3-1	Configuration Diagram of D/A converter	1090
Figure 45.3-2	Register List	1090
Figure 45.5-1	The operations of the D/A converter	1092
Figure 46.2-1	Block diagram of the Alarm Comparator	1097
Figure 46.3-1	Alarm Comparator Control/Status Register (ACSR0, ACSR1)	1098
Figure 46.3-2	Structure of Alarm comparator control/status register	1098
Figure 47.1-1	Block Diagram of LCD Controller	1101
Figure 47.3-1	Configuration Diagram of LCD Controller	1102
Figure 47.3-2	Register List	1103
Figure 47.4-1	Correspondence between VRAM and Common/Segment Pins	1108
Figure 47.5-1	1/2 Duty Cycle Output Waveforms	1111
Figure 47.5-2	1/3 Duty Cycle Output Waveforms	1113
Figure 47.5-3	1/4 Duty Cycle Output Waveforms	1115
Figure 47.7-1	Selection of internal divided resistors	1121
Figure 47.7-2	Selection of external divided resistors	1121
Figure 48.1-1	Block diagram of Clock Monitor	1123
Figure 48.3-1	Configuration Diagram of Clock Monitor	1124
Figure 48.3-2	Register List	1124
Figure 48.5-1	Output waveforms of the Clock Monitor	1127
Figure 49.1-1	Block diagram of the Real-Time Clock	1131
Figure 49.3-1	Configuration of the Real-Time Clock	1132
Figure 49.3-2	Register List of the Real-Time Clock	1132
Figure 49.5-1	Real-time Clock operation	1138
Figure 50.1-1	Block diagram of Sub clock calibration unit	1145
Figure 50.2-1	Timing of the measurement process	1146
Figure 50.4-1	Calibration Unit Control Register (CUCR)	1148
Figure 50.4-2	32kHz/100kHz Timer Data Register (CUTD)	1148
Figure 50.4-3	4MHz Timer Data Register (CUTR1/CUTR2)	1149
Figure 50.4-4	Calibration Unit Control Register (CUCR)	1150
Figure 50.4-5	32kHz / 100kHz Timer Data Register (16 bit) (CUTD)	1151
Figure 50.4-6	4MHz Timer Data Register (24 bits) (CUTR)	1153
Figure 51.2-1	Block diagram of the Low Voltage Reset/Interrupt	1157
Figure 52.2-1	Block diagram of regulator control	1161
Figure 53.2-1	Flow Chart of checking boot conditions on MB91V460A	1166
Figure 53.2-2	Flow Chart of checking boot conditions on flash derivates of MB91460 series	1168
Figure 53.5-1	Normal start of the application	1172
Figure 53.5-2	The application as well as user bootloader 2 can be re-programmed	1172
Figure 53.5-3	Reset during Programming of bootloader 1	1173

Figure 54.3-1	Block Diagram (32bit/64bit flash) internal access.....	1176
Figure 54.3-2	Block Diagram (32bit/64bit flash) external access.....	1176
Figure 54.3-3	Flash sector organisation in CPU access.....	1177
Figure 54.3-4	Flash sector organisation in parallel programming mode.....	1178
Figure 54.7-1	The Writing Sequence Using Write Commands.....	1187
Figure 54.7-2	Chip Erase Sequence Using the Chip erase Command.....	1188
Figure 54.8-1	Example of the flash memory write procedure.....	1198
Figure 54.8-2	Example of the flash memory sector erase procedure.....	1200
Figure 56.6-1	Data flash interrupts.....	1224
Figure 58.2-1	E-Ray block diagram.....	1254
Figure 58.4-1	Structure of communication cycle.....	1339
Figure 58.4-2	Configuration of NIT start and offset correction start.....	1341
Figure 58.4-3	Overall state diagram of E-Ray communication controller.....	1347
Figure 58.4-4	Structure of POC state WAKEUP.....	1352
Figure 58.4-5	Timing of wakeup pattern.....	1354
Figure 58.4-6	State diagram time-triggered startup.....	1356
Figure 58.4-7	FIFO status: empty, not empty, overrun.....	1371
Figure 58.4-8	Host access to Message RAM.....	1374
Figure 58.4-9	Double buffer structure Input Buffer.....	1375
Figure 58.4-10	Swapping of IBCM and IBCR bits.....	1375
Figure 58.4-11	Double buffer structure Output Buffer.....	1377
Figure 58.4-12	Swapping of OBCM and OBCR bits.....	1377
Figure 58.4-13	Access to Transient Buffer RAMs.....	1380
Figure 58.4-14	Configuration example of message buffers in the Message RAM.....	1381
Figure 58.4-15	Parity generation and check.....	1386
Figure 60.2-1	MediaLB Block Diagram.....	1437
Figure 60.3-1	Bit Configuration of the Device Control Configuration Register (DCCR).....	1447
Figure 60.3-2	Bit Configuration of the System Status Configuration Register (SSCR).....	1449
Figure 60.3-3	Bit Configuration of the System Data Configuration Register (SDCR).....	1452
Figure 60.3-4	Bit Configuration of the System Mask Configuration Register (SMCR).....	1453
Figure 60.3-5	Bit Configuration of the Version Control Configuration Register (VCCR).....	1455
Figure 60.3-6	Bit Configuration of the Channel Interrupt Configuration Register (CICR).....	1456
Figure 60.3-7	Bit Configuration of the Channel n Entry Configuration Register (CECRn).....	1457
Figure 60.3-8	Bit Configuration of the Channel n Status Configuration Register (CSCRn).....	1461
Figure 60.3-9	Bit Configuration of the Channel Current Buffer Configuration Register (CCBCRn) During the IO Mode1466	
Figure 60.3-10	Bit Configuration of the Channel n Current Buffer Configuration Register (CCBCRn) during the DMA Mode1466	
Figure 60.3-11	Bit Configuration of the Channel n Next Buffer Configuration Register (CNBCRn) during the IO Mode1468	
Figure 60.3-12	Bit Configuration of the Channel n Next Buffer Configuration Register (CNBCRn) during the DMA Mode1468	
Figure 60.3-13	Bit Configuration of the Local Channel n Buffer Configuration Register (LCBCRn).....	1470
Figure 60.3-14	LCBC Example of the Rn Setting.....	1472
Figure 60.3-15	Bit Configuration of the FIFO Buffer Channel n Address Range Register (BUFARn).....	1473
Figure 60.3-16	Bit Configuration of the Media Stereo Data Setting Register (MSTD).....	1475
Figure 60.3-17	Bit Configuration of the Access Select Register (ASLR).....	1477
Figure 60.3-18	FIFO Bit Configuration of the FIFO Buffer Direction Control Register (BUFDCR).....	1479
Figure 60.3-19	Bit Configuration of the Buffer Interrupt Enable Register (BUFIER).....	1481
Figure 60.3-20	Bit Configuration of the FIFO Buffer Status Register (BUFSR).....	1482
Figure 60.3-21	Bit Configuration of the FIFO Buffer Error Register (BUFER).....	1483
Figure 60.3-22	FIFO Bit configuration of the FIFO buffer Reset Register (BUFRST).....	1484
Figure 60.3-23	FIFO Bit Configuration of the FIFO Buffer Count Register m (BUFCTm).....	1486
Figure 60.3-24	FIFO Bit Configuration of the FIFO Buffer n Control Register (BUFnCR).....	1488

MB91460 Series

Figure 60.3-25	Bit Configuration for FIFO Buffer n Data Register (BUF _n DTR)	1490
Figure 60.4-1	Data Flow When MediaLB is in the IO Mode.....	1491
Figure 60.4-2	Data Flow When MediaLB is in the DMA Mode	1494
Figure 60.4-3	Data Flow for Channel Buffer n with ASLR.ASn[1:0]="00"(n=0,1,...,7)	1504
Figure 60.4-4	Data Flow for Channel Buffer n with ASLR.ASn[1:0]="01" (n=0,1,...,7)	1505
Figure 60.4-5	Data Flow for Channel Buffer n with ASLR.ASn[1:0]="10" (n=0,1,...,7)	1506
Figure 60.4-6	Link between MediaLB and FIFO Buffer Registers	1507
Figure 60.4-7	Relationship between RAM and Channel Buffer	1510
Figure 60.4-8	Setting Example of RAM and Buffer n Control Register (BUF _n CR)	1512
Figure 60.4-9	Data Request Made during Writing by MediaLB	1513
Figure 60.4-10	Data Request Made during Reading by MediaLB	1513
Figure 61.2-1	Bit Configuration of the I ² S Common Control Register (I2SCCR)	1534
Figure 61.2-2	Bit Configuration of the I ² S Rate Setting Register (I2SRSR)	1537
Figure 61.2-3	Bit Configuration of the I ² S Shift Control Register n (I2SSCR _n)	1540
Figure 61.2-4	Bit Configuration of the I ² S Buffer Transfer Count Register n (I2SBTC _n).....	1544
Figure 61.2-5	Bit Configuration of the I ² S Buffer Control Register n (I2SBCR _n)	1546
Figure 61.2-6	Bit Configuration of the Left Data Register n (LTDT _n).....	1548
Figure 61.2-7	Bit Configuration of the Right Data Register n (RTDT _n).....	1549
Figure 61.3-1	Transmission/Reception of the I ² S	1550
Figure 61.3-2	Transmission/Reception of I ² S When I2SCCR:WL=1	1550
Figure 61.3-3	Timings of the MediaLB and I ² S in Single Operation	1551
Figure 61.3-4	Timings of the MediaLB and I ² S in Continuous Operation	1552
Figure 61.5-1	Relationship between the BTC Bits in the I ² S Buffer Control Register and the Interrupt Flags in Transmission (I2SSCR _n :TR="1") (If I2SBCR _n :BTC[3:0]=0100)1554	
Figure 61.5-2	Relationship between the BTC Bits in I ² S Buffer Control Register and the Interrupt Flags in Reception (I2SSCR _n :TR="0") (If I2SBCR _n :BTC[3:0]=0100)1555	
Figure 62.3-1	Block Diagram of USB Mini-host	1559
Figure 62.4-1	Host Control Register Low (HCNTL _n)	1561
Figure 62.4-2	Host Control Register High (HCNTH _n).....	1562
Figure 62.4-3	Bit Configuration of Host Interrupt Register (HIRQ _n)	1564
Figure 62.4-4	Bit Description of the Host Error Status Register (HERR _n)	1566
Figure 62.4-5	Bit Configuration of the Host State Status Register (HSTAT _n)	1568
Figure 62.4-6	Bit Configuration of SOF Interruption FRAME Comparison Register (HFCOMP _n)	1570
Figure 62.4-7	Bit Configuration of Retry Timer Setting Register (HRTIMERL _n , HRTIMERM _n , HRTIMERH _n)....	1571
Figure 62.4-8	Bit Configuration of Host address Register (HADR _n)	1572
Figure 62.4-9	Bit Configuration of EOF Setting Register (HEOF _n).....	1573
Figure 62.4-10	Bit Configuration of FRAME Setting Register (HFRAMEN)	1574
Figure 62.4-11	Bit Configuration of Host Token Endpoint Register (HTOKEN _n).....	1575
Figure 62.5-1	Connection detection timing example of the USB device speed	1577
Figure 62.5-2	Reset Timing Example to Device	1578
Figure 62.5-3	Example of Register Setting until Execution of IN/OUT/SETUP Token	1579
Figure 62.5-4	SOF Timing	1579
Figure 62.5-5	Retry Timer Operation (HIRQ _n :SOFIRQ = "0")	1581
Figure 62.5-6	SOF Interrupt.....	1582
Figure 62.5-7	Example of Token Cancel Operation when HCNTH _n :CANCEL bit is "1".	1583
Figure 62.5-8	Example of Token Operation when HCNTH _n :CANCEL bit is "0".	1583
Figure 62.5-9	First example of HIRQ _n :CMPIRQ Bit Set Timing	1584
Figure 62.5-10	Second example of HIRQ _n :CMPIRQ Bit Set Timing (SOF TOKEN)	1584
Figure 62.5-11	Resume Operation by Register (Full Speed Mode).....	1585
Figure 62.5-12	Resume Operation by detecting K-state on UDP and UDM (Full Speed Mode)	1586
Figure 62.5-13	Resume Operation by Device disconnection.....	1586
Figure 62.5-14	Resume Operation by Device Connection	1587
Figure 62.6-1	Flow Chart at IN, OUT, SETUP Token	1588

Figure 62.6-2	Flow Chart of SOF Token	1589
Figure 63.2-1	Block Diagram of USB Function	1592
Figure 63.3-1	UDC Control Register (UDCCn)	1597
Figure 63.3-2	EP0 Control Register (EP0Cn)	1599
Figure 63.3-3	EP1 Control Register (EP1Cn)	1601
Figure 63.3-4	EP2 to EP5 Control register (EP2Cn to EP5Cn)	1602
Figure 63.3-5	Time Stamp register (TMSPn)	1604
Figure 63.3-6	UDC Status register	1605
Figure 63.3-7	UDC Interrupt Enable Register (UDCIEn)	1607
Figure 63.3-8	EP0I Status register (EP0ISn)	1609
Figure 63.3-9	EP0O Status Register (EP0OSn)	1611
Figure 63.3-10	EP1 Status Register (EP1Sn)	1613
Figure 63.3-11	EP2 to EP5 Status Register (EP2Sn to EP5Sn)	1614
Figure 63.3-12	EP0 to EP5 Data Register (EP0DTn to EP5DTn)	1617
The following describes the function of each bit in the EP0 to EP5 data registers (EP0DTn to EP5DTn)		
Figure 63.4-1	Example of Connecting for USB Cable Terminal	1618
Figure 63.4-2	Communication Stages	1619
Figure 63.4-3	USB connection example	1620
Figure 63.4-4	Operation When Detecting a Connection	1620
Figure 63.4-5	Operation When Detecting Disconnection	1621
Figure 63.4-6	Read Command USB function operation	1622
Figure 63.4-7	Write Command USB function operation	1623
Figure 63.4-8	Suspend Operation	1624
Figure 63.4-9	Remote Wake-up Operation	1624
Figure 63.4-10	Wake-up Operation from Host	1625
Figure 63.4-11	OUT Packet Transfer	1625
Figure 63.4-12	IN Packet Transfer	1626
Figure 63.4-13	OUT Direction (HOST \rightarrow Device) Transfer	1627
Figure 63.4-14	IN Direction (Device \rightarrow HOST) Transfer	1627
Figure 63.4-15	NULL Data Transfer Operation	1628
Figure 64.1-1	In the MB91F467E block diagram, this part of the device is called "Always ON Logic"	1631
Figure 64.4-1	Shutdown Control in the external INIT chain:	1643

MB91460 Series

Appendix C List of Tables

Table 1.3-1	Registers, which should not be accessed with Read-Modify-Write	8
Table 1.3-2	Registers, which include status flags (examples)	8
Table 1.5-1	Main terminology	11
Table 2.3-1	Evaluation device lineup MB91V460A and MB91FV460B	25
Table 2.3-2	MB91460 series device lineup	27
Table 2.3-3	MB91460 series resource lineup	29
Table 3.2-1	I/O Map	37
Table 3.3-1	Interrupt Vector Table	113
Table 3.6-1	Pin definitions MB91V460A	126
Table 3.7-1	I/O circuit types MB91V460A	141
Table 3.8-1	MB91FV460B Series Modes	143
Table 3.8-2	General purpose port and resource function IO names	144
Table 3.8-3	User ports in default series mode	146
Table 3.8-4	Multiplexed pins in series mode P/T	158
Table 3.8-5	Multiplexed pins in series mode B	162
Table 3.8-6	Relocated external interrupts in series mode Q	164
Table 3.8-7	Dedicated and special pins	165
Table 3.8-8	Common ground pin numbers	175
Table 3.8-9	Common ground JEDEC numbers	175
Table 3.8-10	Core supply 1.8V pin numbers	175
Table 3.8-11	Core supply 1.8V JEDEC numbers	176
Table 3.8-12	IO ring power pin numbers	176
Table 3.8-13	IO ring power JEDEC numbers	176
Table 3.8-14	DSU4 IO power 3.3V pin numbers	176
Table 3.8-15	DSU4 IO power 3.3V JEDEC numbers	176
Table 3.8-16	External bus power 3V to 5V pin numbers	177
Table 3.8-17	External bus power 3V to 5V JEDEC numbers	177
Table 3.8-18	USB power 3.3V pin numbers	177
Table 3.8-19	USB power 3.3V JEDEC numbers	177
Table 3.8-20	Media LB IO power 3.3V pin numbers	177
Table 3.8-21	Media LB IO power 3.3V JEDEC numbers	177
Table 3.8-22	Stepper motor power 3V to 5V pin numbers	177
Table 3.8-23	Stepper motor power 3V to 5V JEDEC numbers	177
Table 3.8-24	APIX supply pin numbers	178
Table 3.8-25	APIX supply JEDEC numbers	178
Table 3.8-26	APIX ground pin numbers	178
Table 3.8-27	APIX ground JEDEC numbers	178
Table 3.9-1	MB91FV460B I/O circuit type overview	179
Table 3.10-1	Initial behaviour of external bus	193
Table 3.10-2	Pin states of external bus pins in external bus mode	194
Table 3.10-3	Pin states of general purpose ports and external bus pins in single chip mode	196
Table 3.10-4	Pin states of special I/O pins	197
Table 3.10-5	Pin states of MB91FV460B special I/O pins	199
Table 6.6-1	Interrupt Level of EIT	219
Table 6.8-1	Priority Level of Receipt of EIT Triggers and Mask for Other Triggers	220
Table 6.8-2	Execution Sequence of EIT Handlers	220
Table 10.6-1	Settings Required to Change to SLEEP state	259
Table 10.6-2	Settings Required to Change to STOP state	259
Table 11.7-1	FLASH-IF Registers Summary	264
Table 12.4-1	I-Cache Size Setting	277
Table 12.6-1	I-Cache Status in various operating modes	284
Table 12.6-2	I-Cache Entry Update	285

Table 13.4-1	Allowed transitions from CLKS1 and CLKS0.....	295
Table 13.6-1	Settings for Operating at 1/2 of the Main Clock.....	307
Table 13.6-2	Settings for Operating Using the Main PLL	307
Table 13.6-3	Settings for Operating Using the Sub clock.....	307
Table 13.6-4	Settings for Selecting the Division Ratio for the Operating Clocks	307
Table 14.4-1	PLL Interface Register Overview	317
Table 14.6-1	Recommended settings for Main PLL	325
Table 14.6-2	Recommended settings for E-Ray PLL	326
Table 14.6-3	Recommended settings for USB PLL.....	326
Table 14.7-1	Main PLL in STOP state	327
Table 16.2-1	Functional Description of each bit of the Clock Supervisor Control Register	336
Table 16.4-1	Main Clock Supervisor in RTC mode.	351
Table 16.4-2	Sub Clock Supervisor in RTC mode.	351
Table 17.2-1	Function of each bit of the clock modulator control register	358
Table 17.2-2	States of the modulator	360
Table 17.2-3	Function of each bit of the modulation parameter register (CMPR)	361
Table 18.4-1	Cases When the CLKS1 and CLKS0 Bits May or May Not be Modified	368
Table 18.6-1	Settings Required to Specify the Oscillation Stabilization Wait Time	375
Table 18.6-2	Settings Required to Setup an INITX Pin Reset.....	375
Table 19.6-1	Setting Required for the Timebase Timer.....	384
Table 19.6-2	Setting Required for Interrupting the Timebase Timer	384
Table 20.6-1	Setting Required for Using the Watchdog Timer	396
Table 20.6-2	Setting Required for Delaying the Generation of the Watchdog.....	396
Table 20.6-3	Setting Required for Checking the Generation of the Watchdog.....	396
Table 23.6-1	Settings Required for Using the Sub oscillation stabilization timer.....	423
Table 23.6-2	Items Required for Enabling the Sub oscillation stabilization timer Interrupt.....	423
Table 24.5-1	Interrupt level at which a cancel request is generated	436
Table 24.6-1	Setting Required to Use Interrupts	438
Table 24.6-2	Setting that Requires the Setting within Interrupt Processing	438
Table 25.6-1	Setting Required in Order to Use External Interrupts	460
Table 25.7-1	External Interrupts Port Function Register Setting	460
Table 26.2-1	Settings for Transfer Request Sources	470
Table 26.2-2	Settings for Extended Transfer Request Sources	471
Table 26.2-3	Settings for the Transfer Types	479
Table 26.2-4	Settings for Transfer Modes	480
Table 26.2-5	Selection of the Transfer Data Width.....	480
Table 26.2-6	End Codes-1.....	484
Table 26.2-7	End Codes-2.....	484
Table 26.3-1	Specifiable transfer addresses (burst 2-cycle transfer)	492
Table 26.3-2	Specifiable transfer addresses (burst fly-by transfer)	493
Table 26.3-3	Specifiable transfer addresses (demand transfer 2-cycle transfer)	494
Table 3.2-3	Specifiable transfer addresses (demand transfer fly-by transfer)	494
Table 26.3-4	Specifiable transfer addresses (step/block transfer 2-cycle transfer)	494
Table 26.3-5	Specifiable transfer addresses (step/block transfer fly-by transfer).....	495
Table 26.3-6	Stopping due to an Error	501
Table 27.6-1	Setting required for the delayed interrupt generation/clear	519
Table 28.5-1	The Relationship Between the Bit Position and the Value to be Returned (Decimal Notation).....	526
Table 28.6-1	Settings Required for “Zero” Position Detection	527
Table 28.6-2	Setting Required for Using “One” Position Detection	527
Table 28.6-3	Setting Required for Using Changing Position Detection	527
Table 29.3-1	EDSU Registers Summary	534
Table 29.3-2	BD Coding for Match on Start/Endpoint or Range.....	545
Table 29.3-3	Relationship of BCR, BAD and BIRQ registers	546
Table 29.4-1	Instruction Break Detection Status Bits (BD).....	557

MB91460 Series

Table 29.4-2	Operand Break Detection Status Bits (BD)	558
Table 29.4-3	Operand size and operand address relations	558
Table 29.4-4	Data size and data address relations	560
Table 29.4-5	Operand address and data value break combinations	561
Table 29.4-6	Comparator Type Configuration	561
Table 29.4-7	Meaning of the permission config bits	561
Table 29.4-8	Interrupt numbers and vectors of break factors	562
Table 30.1-1	I/O port functions MB91V460A	565
Table 30.3-1	I/O cell list MB91V460A	591
Table 30.4-1	Function of PORTEN.ADCHE bit	596
Table 30.4-2	PILR settings MB91Fxxx and MB91FV460B (GP00 - GP35, GP37, GP40)	646
Table 30.4-3	PILR settings MB91Fxxx and MB91FV460B (GP36[7:5])	646
Table 30.4-4	PILR settings of MB91Fxxx and MB91FV460B (GP36[4:0], GP38, GP39)	646
Table 30.4-5	PILR settings of the MB91V460A (GP00 - GP13)	647
Table 30.4-6	PILR settings of the MB91V460A (GP14 - GP35)	647
Table 30.4-7	PILR register list	647
Table 30.4-8	EPILR register list	648
Table 30.4-9	PPER register list	650
Table 30.4-10	PPCR register list	651
Table 30.4-11	PODR register list	653
Table 31.2-1	Area Size Settings	662
Table 31.2-2	Setting of the Data Bus Width of Each Chip Select Area	663
Table 31.2-3	Setting of the Maximum Burst Length of Each Chip Select	663
Table 31.2-4	Access Type Settings for Each Chip Select Area	665
Table 31.2-5	Settings for the Number of Auto-Wait Cycles (During First Access)	669
Table 31.2-6	Settings for the Number of Auto-Wait Cycles (During Burst Access)	669
Table 31.2-7	Settings of the Idle Cycle	669
Table 31.2-8	Settings for the Number of Write Recovery Cycles	670
Table 31.2-9	Access Type Settings (TYP3 - TYP0 Bits)	672
Table 31.2-10	Setting the Number of Cycles from RAS Output to CAS Output	672
Table 31.2-11	CAS Latency Setting	673
Table 31.2-12	Read - to - write cycle	673
Table 31.2-13	Write recovery cycle	673
Table 31.2-14	RAS active time	674
Table 31.2-15	RAS precharge cycle	674
Table 31.2-16	Access type settings (TYP3 to TYP0 bits)	675
Table 31.2-17	Settings for the page size of SDRAM	675
Table 31.2-18	Settings for burst write	675
Table 31.2-19	settings for bank number	676
Table 31.2-20	Settings for the number of active banks	676
Table 31.2-21	Access type settings (TYP3 to TYP0 bits)	677
Table 31.2-22	Settings for the Number of I/O Idle Cycles	678
Table 31.2-23	Settings for the Number of I/O Wait Cycles	679
Table 31.2-24	CSXn Corresponding to the Chip Select Enable Bits	680
Table 31.2-25	Settings for Wait Cycle Reduction	682
Table 31.2-26	Settings for self - refresh control	683
Table 31.2-27	Function of refresh counter startup control	683
Table 31.2-28	Function of burst refresh control	684
Table 31.2-29	Number of times to refresh	684
Table 31.2-30	Function of power - on control	685
Table 31.2-31	Settings for the refresh cycle (tRC)	685
Table 31.9-1	SDRAM/FCRAM Interface to SDRAM/FCRAM Connection Table	729
Table 32.1-1	USART functions	751
Table 32.1-2	USART operation modes	752

Table 32.1-3	Mode Bit Setting	753
Table 32.1-4	USART interrupts	753
Table 32.3-1	USART04 Pins	758
Table 32.4-1	USART04 Registers	759
Table 32.4-2	Functions of each bit of control register 04 (SCR04)	761
Table 32.4-3	* Read/Write options of AD-Bit	761
Table 32.4-4	Bit function of the Serial Mode register 04 (SMR04)	763
Table 32.4-5	Functions of each bit of status register 04 (SSR04)	765
Table 32.4-6	Function of each bit of the Extended Status/Control Register (ESCR4)	768
Table 32.4-7	* Description of the interaction of SOPE and SIOP:	768
Table 32.4-8	Function of each bit of the Extended Communication Control Register (ECCR04)	770
Table 32.4-9	Functions of each bit of fifo control register (FCR4)	773
Table 32.4-10	Functions of each bit of FIFO status Register	775
Table 32.5-1	USART Interrupts	776
Table 32.6-1	Suggested Baud Rates and reload values at different CLKP speeds.	784
Table 32.7-1	USART operation mode	787
Table 32.7-2	LIN break length	793
Table 32.7-3	Selection of the master-slave communication function	797
Table 33.2-1	I2C Prescaler Settings	824
Table 33.2-2	I2C SCL frequency versus CLKP settings	824
Table 34.2-1	CAN Register Summary	833
Table 34.2-2	Additional flags when more than 32 message buffers exist	857
Table 34.2-3	Additional flags when more than 32 message buffers exist	859
Table 34.2-4	Additional flags when more than 32 message buffers exist	860
Table 34.2-5	Additional flags when more than 32 message buffers exist	861
Table 34.4-1	Parameters of the CAN Bit Time	874
Table 35.7-1	Setting Required in Order to Use the Free-Run Timer	884
Table 35.7-2	Setting Required to Enable the Free-Run Timer Interrupt	884
Table 35.7-3	Setting Required to Stop the Free-Run Timer	884
Table 36.6-1	Settings Necessary for Using Input Capture	898
Table 36.6-2	Required Settings for ICU Interrupt	898
Table 37.6-1	Settings Necessary for Using Output Compare	912
Table 37.6-2	Item Necessary to Clear the Free-Run Timer upon Compare-match.	912
Table 37.6-3	Item Necessary for Performing Interrupts	912
Table 38.6-1	Settings necessary for internal clock operation	934
Table 38.6-2	Settings necessary for external event operation	934
Table 38.6-4	Settings Necessary for Stopping the Reload Timer	935
Table 38.6-3	Items Necessary for Performing Reload Timer Interrupts	935
Table 38.7-1	TMCSR.CSL[2:0]	936
Table 39.6-1	Settings Needed to Start the PPG	961
Table 39.6-2	Settings Needed to Stop the PPG	961
Table 39.6-3	Settings Needed to Clamp the Output Level	961
Table 39.6-4	Settings Needed to Implement PPG Interrupts	961
Table 40.2-1	CSL Bit Clock Source Settings	979
Table 41.6-1	Required Settings to Run Up/Down Counter in Timer Mode	1007
Table 41.6-2	Required Settings to Run Up/Down Counter in Up/Down Count Mode	1007
Table 41.6-3	Required Settings to Run Up/Down Counter in Phase Difference Count Mode	1008
Table 41.6-4	Required Settings for Up/Down Counter Interrupt	1008
Table 41.6-5	Required Settings to Deactivate Up/Down Counter	1008
Table 42.3-1	Sound generator clock selection	1020
Table 43.2-1	SMC clock selection	1028
Table 43.2-2	SMC output level selection	1033
Table 43.3-1	Selection of Motor Drive Signals and Setting of PWM1 and PWM2 Selection Registers	1035
Table 44.1-1	A/D converter feature list	1037

MB91460 Series

Table 44.3-1	A/D Converter register list	1040
Table 44.3-2	A/D converter pause and resume	1045
Table 44.3-3	A/D Converter ACHMD functionality	1049
Table 44.6-1	A/D Converter operation modes	1058
Table 44.6-2	A/D Continuous mode versus Scan mode	1062
Table 44.7-1	ADC interrupt level registers and vectors	1065
Table 44.9-1	Settings needed to use A/D - Single Shot Conversion Mode	1068
Table 44.9-2	Settings needed to use A/D - Continuous Conversion Mode	1068
Table 44.9-3	Forcing A/D operations to Stop	1069
Table 44.9-4	Items needed to enable A/D Interrupts	1069
Table 44.10-1	A/D Conversion mode selection.	1070
Table 44.10-2	A/D Converter resolution setting (Common result register ADxCR)	1070
Table 44.10-3	A/D Converter resolution setting (Channel result registers ADCxDn)	1070
Table 44.10-4	A/D Conversion analog channel enable with PFR/EPFR	1071
Table 44.13-1	Range Comparator interrupt conditions	1082
Table 44.14-1	Range Comparator Setup Example	1084
Table 44.15-1	Range Comparator interrupt level registers and vectors	1086
Table 45.6-1	Settings Needed to Use D/A	1093
Table 45.6-2	Settings Needed to Stop D/A Output	1093
Table 45.8-1	D/A conversion output voltages in 10-bit mode	1095
Table 45.8-2	D/A conversion output voltages in 8-bit mode	1095
Table 45.8-3	D/A conversion speed	1095
Table 46.4-1	Valid interrupt events	1099
Table 46.4-2	Alarm Comparator power down modes	1100
Table 47.5-1	Example of Data Memory Contents for display	1110
Table 47.5-2	Example of Data Memory Contents for Display	1112
Table 47.5-3	Example of Data Memory Contents for Display	1114
Table 47.6-1	Required Setting to Use LCD	1116
Table 47.6-2	Required Setting to Disable LCD display	1116
Table 47.6-3	Required Setting to Deactivate LCD	1116
Table 47.6-4	Required Setting to Enable LCD Display in Sub-STOP state	1116
Table 47.7-1	1/2 duty cycle	1119
Table 47.7-2	1/3 duty cycle	1119
Table 47.7-3	1/4 duty cycle	1119
Table 48.6-1	Settings for Using Clock Monitor	1128
Table 49.6-1	Required Settings to Run Real-time Clock	1140
Table 49.6-2	Required Settings to Know Time	1140
Table 49.6-3	Required Settings to Stop Real-Time Clock	1140
Table 49.6-4	Required Settings to Generate Interrupts to Real-time Clock	1140
Table 49.6-5	Required Settings to select clock source to Real-time Clock	1140
Table 50.3-1	Maximum operation frequencies	1147
Table 50.3-2	Example of valid clock ratios which fulfill requirements 1 and 2	1147
Table 50.4-1	Calibration Unit registers	1149
Table 50.4-2	32kHz : Ideal measurement results depending on measurement duration	1152
Table 50.4-3	100kHz : Ideal measurement results depending on measurement duration	1152
Table 50.5-1	Ideal measurement results (CUTR) with 32.768kHz and 4.0MHz oscillators	1154
Table 50.5-2	Ideal measurement results (CUTR) with 100kHz and 4.0MHz oscillators	1154
Table 53.2-1	Boot security vector address ranges	1167
Table 53.2-2	Boot loader commands	1169
Table 53.3-1	Reset Source Register (RSRR) data storage addresses	1170
Table 53.6-1	List of devices with Chip-ID at location 0xBFF0	1173
Table 53.6-2	List of Devices with Chip-ID at location 0xBFF4	1174
Table 54.4-1	Main Flash Register Overview	1180
Table 54.6-1	MB91460 series CPU flash access modes	1183

Table 54.7-1	Flash command sequences	1185
Table 54.7-2	Hardware sequence flag format	1189
Table 54.7-3	Hardware sequence flags	1189
Table 54.7-4	Hardware sequence flag functions	1190
Table 55.3-1	Explanation of the bits in the Flash Security Vector FSV1[31:16]:	1206
Table 55.3-2	Explanation of the bits in the Flash Security Vector FSV1[15:0] (valid for MB91F467Dx): ...	1207
Table 55.3-3	Explanation of the bits in the Flash Security Vector FSV2[31:0] (valid for MB91F467Dx): ...	1208
Table 56.4-1	Data flash command sequences	1216
Table 56.4-2	Data flash hardware sequence flags	1217
Table 58.3-1	Assignment of message buffers	1256
Table 58.3-2	E-Ray register map	1260
Table 58.3-3	Reference to CHI Host command summary from FlexRay protocol specification	1285
Table 58.3-4	Assignment of data bytes to network management vector	1310
Table 58.3-5	Coding for releases	1323
Table 58.4-1	Error modes of the POC (degradation model)	1345
Table 58.4-2	State transitions of E-Ray overall state machine	1348
Table 58.4-3	State transitions WAKEUP	1352
Table 58.4-4	Definition of cycle set	1364
Table 58.4-5	Examples for valid cycle sets	1364
Table 58.4-6	Channel filtering configuration	1365
Table 58.4-7	Scan of Message RAM	1372
Table 58.4-8	Assignment of IBCM bits	1376
Table 58.4-9	Assignment of IBCR bits	1376
Table 58.4-10	Assignment of OBCM bits	1379
Table 58.4-11	Assignment of OBCR bits	1379
Table 58.4-12	Header section of a message buffer in the Message RAM	1382
Table 58.4-13	Example for structure of the data partition in the Message RAM	1385
Table 58.4-14	Module interrupt flags and interrupt line enable	1391
Table 58.5-1	Register bit overview	1403
Table 58.5-2	FlexRay configuration parameters	1405
Table 60.1-1	Term Descriptions	1436
Table 60.3-1	List of the MediaLB Registers	1440
Table 60.3-2	MediaLB Bit Allocation	1445
Table 60.3-3	Functional Descriptions on the Device Control Configuration Register (DCCR) Bits	1447
Table 60.3-4	Functional Descriptions on the System Status Configuration Register (SSCR) Bits	1450
Table 60.3-5	Functional Descriptions on the System Data Configuration Register (SDCR) Bits	1452
Table 60.3-6	Functional Descriptions on the System Mask Configuration Register (SMCR) Bits	1454
Table 60.3-7	Functional Descriptions on the Version Control Configuration Register (VCCR) Bits	1455
Table 60.3-8	Functional Descriptions on the Channel Interrupt Configuration Register (CICR) Bits	1456
Table 60.3-9	Functional Descriptions on the Channel n Entry Configuration Register (CECRn) Bits	1457
Table 60.3-10	Functional Descriptions on the Channel n Status Configuration Register (CSCRn) Bits	1462
Table 60.3-11	Functional Descriptions on the Channel n Current Buffer Configuration Register (CCBCRn) Bits during the IO Mode1466	
Table 60.3-12	Functional Descriptions on the Channel Current Buffer Configuration Register (CCBCRn) Bits during the DMA Mode1467	
Table 60.3-13	Functional Descriptions on the Channel n Next Buffer Configuration Register (CNBCRn) Bits during the IO Mode1468	
Table 60.3-14	Functional Descriptions on the Channel n Next Buffer Configuration Register (CNBCRn) Bits during the DMA Mode1469	
Table 60.3-15	Functional Descriptions on the Local Channel n Buffer Configuration Register (LCBCRn) Bits ... 1471	
Table 60.3-16	Functional Descriptions on the FIFO Buffer Channel n Address Range Register (BUFARn) Bits 1474	
Table 60.3-17	Functional Descriptions on the Media Stereo Data Setting Register (MSTD) Bits	1476

MB91460 Series

Table 60.3-18	Functional Descriptions on the Access Select Register Bits	1478
Table 60.3-19	Functional Descriptions on the FIFO Buffer Direction Control Register Bits	1480
Table 60.3-20	Functional Descriptions on the FIFO Buffer Interrupt Enable Register (BUFIER) Bits	1481
Table 60.3-21	Functional Descriptions on the FIFO Buffer Status Register (BUFSR) Bits	1483
Table 60.3-22	Functional Descriptions on the FIFO Buffer Error Register (BUFER) Bits	1484
Table 60.3-23	Functional Descriptions on the FIFO Buffer Reset Register (BUFRST) Bits	1485
Table 60.3-24	Functional Descriptions on the FIFO buffer Count Register m (BUFCTm) Bits	1487
Table 60.3-25	Functional Descriptions on the FIFO Buffer n Control Register (BUFnCR) Bits	1489
Table 60.3-26	Functional Description of Each Bit in FIFO Buffer n Data Register (BUFnDTR)	1490
Table 61.2-1	List of the I ² S Registers	1531
Table 61.2-2	I ² S Bit Allocations	1533
Table 61.2-3	Functional Descriptions on the I ² S Common Control Register Bits	1535
Table 61.2-4	Functional Descriptions on the I ² S Rate Setting Register Bits	1538
Table 61.2-5	Functional Descriptions on the I ² S Shift Control Register n (I2SSCRn) Bits	1541
Table 61.2-6	Functional Descriptions on the I ² S Buffer Transfer Count Register n (I2SBTn) Bits	1545
Table 61.2-7	Functional Descriptions on the I ² S Buffer Control Register n (I2SBCRn) Bits	1547
Table 61.2-8	Functional Descriptions on the Left Data Register n (LTDTn) Bits	1548
Table 61.2-9	Functional Descriptions on the Right Data Register n (RTDTn) Bits	1549
Table 62.4-1	Registers of USB Mini-host	1560
Table 62.4-2	Functional Description of each bit of the HCNtn registers:	1562
Table 62.4-3	Functional Description of each bit in the HIRQn registers:	1565
Table 62.4-4	Functional Description of each bit of the HERRn registers:	1567
Table 62.4-5	Functional Description of each bit of the HSTATEn registers:	1569
Table 62.4-6	Functional Description of each bit of the HFCOMPn registers:	1570
Table 62.4-7	Functional Description of each bit of the HRTIMERLn, HRTIMERMn, HRTIMERHn registers:	1571
Table 62.4-8	Functional Description of each bit of the HADRn registers:	1572
Table 62.4-9	Functional Description of each bit of the HEOFn registers:	1573
Table 62.4-10	Functional Description of each bit of the HFRAMEn registers:	1574
Table 62.4-11	Functional Description of each bit of the HTOKENn registers:	1575
Table 62.4-12	Token Setting	1576
Table 63.3-1	Register List	1593
Table 63.3-2	Registers of USB function	1593
Table 63.3-3	Functional Description of each bit of the UDC control register (UDCC).	1598
Table 63.3-4	Functional Description of each bit of the EP0 Control Register (EP0Cn).	1599
Table 63.3-5	Functional Description of each bit of the EP1 to EP5 control register (EP1Cn to EP5Cn). ...	1603
Table 63.3-6	Functional Description of each bit of the time stamp register (TMSPn).	1604
Table 63.3-7	Functional Description of each bit of the UDC status register (UDCSn):	1606
Table 63.3-8	Functional Description of each bit of the UDC Interruption Enable Register (UDCIEn).	1608
Table 63.3-9	Functional Description of each bit of the EP0I Status Register (EP0ISn).	1610
Table 63.3-10	Functional Description of each bit of the EP0O Status Register (EP0OSn).	1612
Table 63.3-11	Functional Description of each bit of the EP1 to EP5 Status Register (EP1Sn to EP5Sn). ..	1615
Table 63.3-12	Functional Description of each bit of the EP0 to EP5 Data Register (EP0DTn to EP5DTn). ..	1617
Table 64.3-1	External interrupts which can be set as recovery sources	1633
Table 64.3-2	Source level control bits of the external shutdown recovery interrupt pins	1637
Table 64.4-1	Port Input Level Registers of the external shutdown recovery interrupt pins	1641
Table 64.4-2	Registers and flags for determination of the reset source	1643
Table 64.4-3	Registers which are not initialized by Shutdown Recovery:	1644
Table 64.4-4	Pin states in ShutDown mode	1645
Table A.1-1	Zero Extension and Minus Extension Values of 4-bit Immediate Data	1650
Table A.1-2	Specification of General-purpose Register based on Rj/Ri	1652
Table A.1-3	Specification of Dedicated Register based on Rs	1652
Table A.1-4	Correspondence between reglist of LDM0, LDM1 Instruction and General-purpose Register	1653
Table A.1-5	Correspondence between reglist of STM0, STM1 Instruction and General-purpose Register	1653

Table A.2-1	Add/Subtract Instructions (10 Instructions).....	1657
Table A.2-2	Compare Calculation Instructions (3 Instructions).....	1658
Table A.2-3	Logical Calculation Instructions (12 Instructions)	1658
Table A.2-4	Bit Operation Instructions (8 Instructions)	1658
Table A.2-6	Shift Instructions (9 Instructions)	1659
Table A.2-7	Immediate Data Transfer Instructions (3 Instructions).....	1659
Table A.2-5	Multiply/ Divide Instructions (10 Instructions)	1659
Table A.2-8	Memory Load Instructions (13 Instructions)	1660
Table A.2-10	Inter-Register Transfer Instructions/Dedicated Register Transfer Instructions (5 Instructions)	1661
Table A.2-9	Memory Store Instructions (13 Instructions).....	1661
Table A.2-11	Non-delayed Branching Instructions (23 Instructions).....	1662
Table A.2-12	Delayed Branching Instructions (20 Instructions)	1663
Table A.2-13	Direct Addressing Instructions (14 Instructions)	1664
Table A.2-14	Other Instructions (16 Instructions)	1665
Table A.3-1	Instruction Map	1667
Table A.4-1	Instruction Map of Instruction Format TYPE-E	1668
Table D.1-1	Changes in revision 3.0	1691
Table D.2-1	Added chapters	1691
Table D.2-2	Chapters which got major changes	1692
Table D.2-3	Changes based on MB91460 hardware manual errata	1693
Table D.2-4	Misc Changes.....	1695

MB91460 Series

Appendix D Revision History

D.1 Changes in This Revision

This appendix lists the changes done in hardware manual revision 3.0 (January 2011) versus revision 2.0 (November 2010).

Table D.1-1 Changes in revision 3.0

Page	Chapter or Section	Changes
36	Chapter 3 MB91460 Series Basic Information 3.2. I/O Map	Address 000D59 _H is DDR25 instead of DDR24
441 457	Chapter 25 External Interrupt and NMI 25.1. Overview 25.5. Operation	Added description of non-maskable interrupt (NMI)
1251	Chapter 58 E-Ray	Removed section 58.3 Generic Interface; Renamed eray_bclk into CLKB, eray_sclk into ERAY_SCLK (same naming as in PLL interface chapter); Corrected the interrupt line names: eray_int1 --> FlexRay Channel 1 eray_int0 --> FlexRay Channel 0 eray_tint0 --> FlexRay Timer interrupt 0 eray_tint1 --> FlexRay Timer interrupt 1 (same as listed in Interrupt Vector Table)
1256	Chapter 58 E-Ray 58.3.1 Register Map	Added the customer register list, removed the test registers
1261	Chapter 58 E-Ray 58.3.2 Customer Registers	Added the customer register explanation
1264	Chapter 58 E-Ray 58.3.3 Special Registers	Removed the test registers

D.2 Changes in Preceding Revisions

This appendix lists the changes done in hardware manual revision 2.0 (November 2010) versus revision 1.21 (February 2008)

Table D.2-1 lists the chapters which are newly added.

Table D.2-1 Added chapters

Page	Chapter	Added because...
1213	Chapter 56 Embedded Data Flash	used in MB01FV460B and MB91F467P
1231	Chapter 57 APIX® Controller	used in MB91F467S
1251	Chapter 58 E-Ray	used in MB91F465X

Table D.2-1 Added chapters (continued)

Page	Chapter	Added because...
1407	Chapter 59 MediaLB Clock Generation/Bus Interface	used in MB91F467M
1435	Chapter 60 MediaLB (Media Local Bus interface)	
1531	Chapter 61 I ² S (Inter-Integrated Circuit Sound)	
1557	Chapter 62 USB Mini-Host	used in MB01FV460B
1591	Chapter 63 USB Function	used in MB01FV460B
1631	Chapter 64 ShutDown Control	used in MB01F467E
1649	Appendix A Instruction Lists	the author of this book uses and likes it

The following table D.2-2 lists chapters and sections which got major or heavy changes. Not all the changes there are marked with change bars.

Table D.2-2 Chapters which got major changes

Page	Chapter or Section	Changes
1	Chapter 1 Introduction, Handling and Precautions	Added new Handling Precautions , Software Precautions , Notes on Debugger .
17	Chapter 2 MB91460 Rev.A/ Rev.B Overview	Added MB91FV460B and its features. Added device overview tables in MB91460 Series Product Lineup , added Debug Support Features of MB91FV460B.
35	Chapter 3 MB91460 Series Basic Information	3.1. Memory Map : Added MB91FV460B 3.2. I/O Map : Added all new addresses of MB91FV460B 3.4. Package : Added MB91FV460B 3.8. Pin Definitions MB91FV460B : Added this section 3.9. I/O Circuit Types MB91FV460B : Added this section 3.10. Pin State Table : Total update of this section
237	Chapter 9 Reset	There is a separation into Hardware Watchdog / Clock Supervisor reset (INIT: Settings Initialization Reset) and Software Watchdog Reset (INIT: Settings Initialization Reset) . Additionally, the reset chain is explained and wakeup from the ShutDown mode is added.
275	Chapter 12 External Bus Instruction Cache (I-Cache)	There are no logical changes, but this chapter is totally updated for better understanding, similar to MB91460M series hardware manual. Additionally, the cache is called "I-Cache" to better distinguish between the flash cache ("F-Cache") and this cache.
313	Chapter 14 PLL Interfaces (Main PLL, E-Ray, USB)	This chapter got a general update for better understanding. For MB91FV460B, the PLLs for E-Ray and USB are added.
333	Chapter 16 Clock Supervisor	Totally updated the chapter Clock Supervisor and added the new feature of switching back from RC to MAIN or SUB clock by clearing MM/SM bits.
399	Chapter 21 Hardware Watchdog Timer	General update for better understanding; added functionality of STP_RUN (run in STOP state).

MB91460 Series

Table D.2-2 Chapters which got major changes (continued)

Page	Chapter or Section	Changes
441	Chapter 25 External Interrupt	For MB91FV460B, interrupts 16-31 have been added. The Interrupt Re-location features are explained, as well as the behaviour regarding Level and Edge Detection .
1037	Chapter 44 A/D Converter (ADC) / Range Comparator	This chapter got a general update because of the ADCs in the MB91460 series devices are equipped differently. For some devices there is a Range Comparator added. It is explained how to enable the analog input lines and how the data protection feature works. Corrections: The Operation Modes of A/D Converter , especially the waveform drawings, have been corrected. The calculation formulas of the sampling time $T_{\text{samp}}[\text{min}]$ as well as the comparison time T_{comp} are updated.
1175	Chapter 54 Flash Memory	The sequence of the sections was changed for better understanding. An example of Flash Memory Sector Organisation was inserted and the Flash Access Mode Switching is explained more in detail. The description of Auto Program Algorithms and Hardware Sequence Flags is updated for better understanding and correctness.

The changes in table [D.2-3](#) resulted of items in the hardware manual errata.

Table D.2-3 Changes based on MB91460 hardware manual errata

Page	Errata Ref. Nr.	Description/Correction
1182	HWM91460002	Timing requirement of the PARALLEL programming mode of the flash security: Added a note in section 54.5.2 Flash Memory Mode (Parallel Programmer Mode) .
395	HWM91460005	Watchdog Timer: Retriggered /cleared automatically by DMA (MB91V460A, MB91461): Added a note in Postponing the Generation of a Watchdog Reset
1173	HWM91460006	Chip IDs: Added new section 53.6. Chip-ID containing all chip ID tables.
773	HWM91460007	USART FIFO FIFO Control Register (FCR04) Rx triggerlevel: The note below Table 32.4-9 Functions of each bit of fifo control register (FCR4) was corrected
356	HWM91460008	MB91V460A: Input frequency for clock modulator: Added a note in section 17.2.1 Clock modulator registers
333	HWM91460009, HWM91460033	RC clock run mode after clock supervisor reset cannot be left with hardware watchdog reset + Corrected Chapter "Clock Supervisor": Totally updated the chapter Chapter 16 Clock Supervisor and added the new feature of switching back from RC to MAIN or SUB clock by clearing MM/SM bits.
1183	HWM91460010 HWM91460024	Flash access mode switching: Updated the section 54.6. Flash Access Mode Switching . The start addresses in the Boot ROM are also listed here.
247 248	HWM91460011	Watchdog reset: There is a separation now into Hardware Watchdog / Clock Supervisor reset (INIT: Settings Initialization Reset) and Software Watchdog Reset (INIT: Settings Initialization Reset) .
765	HWM91460012	USART; Status bits of register SSR04 (TDRE): Corrected in Table 32.4-5 Functions of each bit of status register 04 (SSR04) .

Table D.2-3 Changes based on MB91460 hardware manual errata (continued)

Page	Errata Ref. Nr.	Description/Correction
1052	HWM91460013, HWM91460034	A/D Converter; Sampling Timer Setting Register (ADCT/ADxCT1) : The badly formatted examples have been removed in general update of ADC chapter
1120	HWM91460014	LCD: Enabling LCD even in the Sub-STOP state: Corrected in section 47.7.7 How to enable LCD display even in the Sub-STOP state .
1107 1122	HWM91460015	LCD: "Cautions" Corrections added: Corrected in section 47.4.3 VRAM: Data Memory for Display and 47.8. Caution .
1116	HWM91460016	LCD: "Settings" Corrections added: Corrected in Table 47.6-4 Required Setting to Enable LCD Display in Sub-STOP state . Added notes that the LCD of MB91V460A cannot run in any kind of STOP state.
1104	HWM91460017	LCD: Control Register: Note added at end of section 47.4.1 LCR0: LCDC Control Register 0 that LCD can also operate on RC clock.
407	HWM91460018	Main Clock Oscillation Stability Wait Timer: Corrected in section 22.4.1 OSCRH: Control Register for the Main Clock Oscillation Stability Wait Timer
577	HWM91460019	Corrected I/O Signal description for Pins 16_0 to 16_6 (for MB91V460A): Corrected the table on page 577 .
899	HWM91460020	ICU interrupts: Corrected edge polarity description: Corrected in section 36.7.1 What are the types of active edge polarity for external input, and how to select them? .
769	HWM91460021	Correction of note regarding bits of the Extended Communication Control Register (ECCR04): Figure 32.4-6 Configuration of the Extended Communication Control Register (ECCR04) .
1170	HWM91460022	Registers modified by BootROM: Added Table 53.3-1 Reset Source Register (RSRR) data storage addresses with the correct data of all MB91460 series.
537	HWM91460025, HWM91460026	Corrected EDSU Control Register (BCTRL) description: Changed "register" into "bit" in section 29.3.2 Explanations of Registers .
546	HWM91460027	EDSU; Corrected relationship of BCR, BAD and BIRQ register description: Corrected Table 29.3-3 Relationship of BCR, BAD and BIRQ registers .
558	HWM91460028	EDSU; Corrected operand size and operand address relations description: Corrected Table 29.4-3 Operand size and operand address relations .
560	HWM91460029	EDSU; Corrected data value break description Corrected Table 29.4-4 Data size and data address relations .
537	HWM91460030	EDSU; Note regarding Instruction Address Capture Register (BIAC): Corrected in section 29.3.2 Explanations of Registers .
770	HWM91460031	Extended Communication Control Register (ECCR04); Added function description regarding Start/Stop bit mode enable(bit3): Corrected in Table 32.4-8 Function of each bit of the Extended Communication Control Register (ECCR04)
289	HWM91460032	Corrected block diagram of clock distribution: Corrected in Figure 13.1-1 Block Diagram of Clock Distribution
419	HWM91460035	The Sub Oscillation Stabilization Timer operation enable bit WPCRH.WEN is not available: Corrected in section 23.4.1 WPCRH: Sub oscillation stabilization timer Control Register

MB91460 Series

Table D.2-3 Changes based on MB91460 hardware manual errata (continued)

Page	Errata Ref. Nr.	Description/Correction
414, 426	HWM91460036	Oscillation Stabilization Timer clear by WCL="0": Added sections to "Caution": 22.8.1 Timer Clear by Writing WCL="0" for Main Osc. Stabilization Timer, 23.8.1 Timer Clear by Writing WCL="0" for Sub Osc. Stabilization Timer, and links to these sections at WCL bit descriptions.
803	HWM91460037	SCK pulses by UPCL in master mode 2 when mark level="0": Added the workaround to "Software reset of UART" under 32.8. Notes on using USART .

The following table [D.2-4](#) lists miscellaneous changes.

Table D.2-4 Misc Changes

Page	Chapter or Section	Changes
2	1.2. Handling Precautions	Added the latest handling precautions
7	1.3. Software Precautions	Added precautions about read-modify-write access to certain registers. Corrected that LIN-USRAT SMR register can be written using R-M-W but ECCR cannot. Added a lot of other precautions.
10	1.4. Notes on Debugger	This chapter was updated for better understanding and to match the information given in the datasheets.
231	Chapter 8 Device State Transition	Added the ShutDown state to several descriptions, added section 8.3.4 SHUT DOWN .
237	Chapter 9 Reset	Section 9.4.4 Mode Vector : MODR can be written on the EVA devices (V460A + B) in emulation mode.
253	Chapter 10 Standby	Added ShutDown state (partly)
263	Chapter 11 Memory Controller, Flash and F-Cache	The cache connected to the Main Flash memory is called "Flash Cache" or "F-Cache" now. Renamed MB91V460 into MB91V460A, added more links to Flash access mode switching explanation; Updated the remarks in section 11.8.4 FLASH Memory Wait Timing Register (FMWT) regarding the device names.
289	Chapter 13 Clock Control	Added the registers OSCC1: Main Oscillator Control Register and OSCC2: Sub Oscillator Control Register ; explained FCI and RFBEN functionality in section Oscillator Configuration ; updated the register list in Figure 13.4-1 Clock Control Register List .
327	Chapter 14 PLL Interfaces (Main PLL, E-Ray, USB)	Corrected that it is possible to run the PLL in STOP state. Added section 14.7.2 Main PLL Operation in STOP State showing the restrictions.
333	Chapter 16 Clock Supervisor	Figure 16.2-1 Configuration Clock Supervisor Control Register (CSVCR) : CSVCR.SCKS "only for single clock devices" corrected into "in single clock devices always 0". Table 16.2-1 Functional Description of each bit of the Clock Supervisor Control Register : Added that modifying SCKS is only allowed in main clock mode.

Table D.2-4 Misc Changes (continued)

Page	Chapter or Section	Changes
502	26.3.10 DMAC Interrupt Control	DMAC Interrupt Control: An interrupt request can be cleared by writing 000B to DSS2 to 0 (end code) of DMACS... --> DMACS register does not exist. --> The DSS[2:0] bits are in DMACB register, bits[2:0]"
655	Chapter 31 External Bus	Corrected the naming mismatches between WRX0, WRX1, WRX2, WRX3 (or WRXn) and WEX (was called WRX sometimes) on several pages; In section 31.6. Burst Access Operation , it was added that: “For first read access, D31-0 is read when MCLKO rises in the cycle in which the wait cycle ended after RDX was asserted.” and “For page read access, D31-0 is read when MCLKO rises in the cycle in which the wait cycle ended after address was changed.” to explain the burst read timing more precisely.
751	Chapter 32 USART (LIN / FIFO)	Added “End of Transmission” interrupt as well as its control signals and flag; introduced the symbolic name for FSR04_NVFD[4:0] bits as “number of valid FIFO data”; updated the descriptions in 32.5. USART Interrupts ; corrected the baudrate prescaler/reload counter to be 16-bit width; updated Table 32.6-1 Suggested Baud Rates and reload values at different CLKP speeds . with higher CLKP speed settings
875	Chapter 35 Free-Run Timer (FRT)	Added FRT8 to FRT11 for MB91FV460B
889	Chapter 36 Input Capture Unit (ICU)	Added ICU8 to ICU9 for MB91FV460B
941	Chapter 39 Programmable Pulse Generator (PPG)	Added PPG16 to PPG31; registers PDUT and PCSR can be read on MB91FV460B, MB91F467P and MB91F467E.
1000	Chapter 41 Up/Down Counter (UDC)	In Figure 41.5-1 Up/Down Counter in Timer Mode CMS[1:0] = “00” , the signal CGSC has been corrected into UDCLR.
1017	Chapter 42 Sound Generator (SG)	Added SGDAD register (SG Device Auto Disable reg) in Figure 42.3-1 Sound Generator Register overview ; explanation of SGDAD in 42.3.4 Device Auto Disable Register (SGDAD)
1044	44.3.4 A/D Control Status Register (ADCS1/ADxCS1)	ADxCS1.STRT - write 1 allowed, write 0 not - strt line is only 1 clock pulse, so read is always 0 - RMW returns 0
1089	Chapter 45 D/A Converter (DAC)	Section 45.4.1 DADR: D/A Data Register : Added info about read values in 8-bit and 10-bit mode.
1097	Chapter 46 Alarm Comparator	ALARM Comp block diagram: “AVDD” is corrected to “AVcc5”
1157	Chapter 51 Low Voltage Reset/Interrupt (Supply Supervisor)	The low voltage detector is also called “Supply Supervisor”
1158	51.3.1 LV Detection Control Registers	LVSEL.LVISEL[3:0] initial settings depending on device
1162	52.3.1 Regulator Control Registers	REGSEL.SUBSEL[3:0] initial settings depending on device

MB91460 Series**Table D.2-4 Misc Changes (continued)**

Page	Chapter or Section	Changes
1209	Chapter 55 Flash Security	In section 55.4.1 Flash Security Control Register , the description of FSCR1.SVF_RDY has been added.

MN705-00001-3v0-E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL
FR60
32-BIT MICROCONTROLLER
MB91460 Series
Hardware Manual

January 14, 2011, the third edition

Published **FUJITSU SEMICONDUCTOR LIMITED**
Edited European Microcontroller Design Centre
Sales Promotion Department
