



---

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.

**F<sup>2</sup>MC-8FX**  
**8-BIT MICROCONTROLLER**  
**MB95110B/M Series**  
**HARDWARE MANUAL**



# F<sup>2</sup>MC-8FX

## 8-BIT MICROCONTROLLER

# MB95110B/M Series

# HARDWARE MANUAL

For the information for microcontroller supports, see the following web site.

This web site includes the "**Customer Design Review Supplement**" which provides the latest cautions on system development and the minimal requirements to be checked to prevent problems before the system development.

<http://edevic.fujitsu.com/micom/en-support/>

FUJITSU MICROELECTRONICS LIMITED



# PREFACE

## ■ The Purpose and Intended Readership of This Manual

Thank you very much for your continued special support for Fujitsu Microelectronics products.

The MB95110B/M series is a line of products developed as general-purpose products in the F<sup>2</sup>MC-8FX series of proprietary 8-bit single-chip microcontrollers applicable as application-specific integrated circuits (ASICs). The MB95110B/M series can be used for a wide range of applications from consumer products including portable devices to industrial equipment.

Intended for engineers who actually develop products using the MB95110B/M series of microcontrollers, this manual describes its functions, features, and operations. You should read through the manual.

For details on individual instructions, refer to the "F<sup>2</sup>MC-8FX Programming Manual".

Note: F<sup>2</sup>MC is the abbreviation of FUJITSU Flexible Microcontroller.

## ■ Trademarks

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

## ■ Sample Programs

Fujitsu provides sample programs free of charge to operate the peripheral resources of the F<sup>2</sup>MC-8FX family of microcontrollers. Feel free to use such sample programs to check the operational specifications and usages of Fujitsu microcontrollers.

Microcontroller support information:

<http://www.fujitsu.com/global/services/microelectronics/product/micom/support/>

Note that sample programs are subject to change without notice. As these pieces of software are offered to show standard operations and usages, evaluate them sufficiently before use with your system. Fujitsu assumes no liability for any damages whatsoever arising out of the use of sample programs.

- 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 MICROELECTRONICS device; FUJITSU MICROELECTRONICS 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 MICROELECTRONICS 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 MICROELECTRONICS or any third party or does FUJITSU MICROELECTRONICS warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU MICROELECTRONICS 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 MICROELECTRONICS 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.

# CONTENTS

<b>CHAPTER 1</b>	<b>DESCRIPTION .....</b>	<b>1</b>
1.1	Feature of MB95110B/M Series .....	2
1.2	Product Lineup of MB95110B/M Series .....	4
1.3	Difference Points among Products and Notes on Selecting a Product .....	8
1.4	Block Diagram of MB95110B/M Series .....	10
1.5	Pin Assignment .....	11
1.6	Package Dimension .....	14
1.7	Pin Description .....	17
1.8	I/O Circuit Type .....	20
<b>CHAPTER 2</b>	<b>HANDLING DEVICES .....</b>	<b>23</b>
2.1	Device Handling Precautions .....	24
<b>CHAPTER 3</b>	<b>MEMORY SPACE .....</b>	<b>29</b>
3.1	Memory Space .....	30
3.1.1	Areas for Specific Applications .....	32
3.2	Memory Map .....	33
<b>CHAPTER 4</b>	<b>MEMORY ACCESS MODE .....</b>	<b>35</b>
4.1	Memory Access Mode .....	36
<b>CHAPTER 5</b>	<b>CPU .....</b>	<b>37</b>
5.1	Dedicated Registers .....	38
5.1.1	Register Bank Pointer (RP) .....	40
5.1.2	Direct Bank Pointer (DP) .....	41
5.1.3	Condition Code Register (CCR) .....	43
5.2	General-purpose Registers .....	45
5.3	Placement of 16-bit Data in Memory .....	47
<b>CHAPTER 6</b>	<b>CLOCK CONTROLLER .....</b>	<b>49</b>
6.1	Overview of Clock Controller .....	50
6.2	Oscillation Stabilization Wait Time .....	56
6.3	System Clock Control Register (SYCC) .....	58
6.4	PLL Control Register (PLLC) .....	60
6.5	Oscillation Stabilization Wait Time Setting Register (WATR) .....	63
6.6	Standby Control Register (STBC) .....	66
6.7	Clock Modes .....	69
6.8	Operations in Low-power Consumption Modes (Standby Modes) .....	75
6.8.1	Notes on Using Standby Mode .....	76
6.8.2	Sleep Mode .....	80
6.8.3	Stop Mode .....	81
6.8.4	Time-base Timer Mode .....	82



6.8.5	Watch Mode .....	83
6.9	Clock Oscillator Circuits .....	84
6.10	Overview of Prescaler .....	86
6.11	Configuration of Prescaler .....	87
6.12	Operating Explanation of Prescaler .....	88
6.13	Notes on Use of Prescaler .....	89
<b>CHAPTER 7</b>	<b>RESET .....</b>	<b>91</b>
7.1	Reset Operation .....	92
7.2	Reset Source Register (RSRR) .....	96
7.3	Notes on Using Reset .....	99
<b>CHAPTER 8</b>	<b>INTERRUPTS .....</b>	<b>101</b>
8.1	Interrupts .....	102
8.1.1	Interrupt Level Setting Registers (ILR0 to ILR5) .....	104
8.1.2	Interrupt Processing Steps .....	105
8.1.3	Nested Interrupts .....	108
8.1.4	Interrupt Processing Time .....	109
8.1.5	Stack Operations During Interrupt Processing .....	110
8.1.6	Interrupt Processing Stack Area .....	111
<b>CHAPTER 9</b>	<b>I/O PORT .....</b>	<b>113</b>
9.1	Overview of I/O Ports .....	114
9.2	Port 0 .....	115
9.2.1	Port 0 Registers .....	117
9.2.2	Operations of Port 0 .....	118
9.3	Port 1 .....	120
9.3.1	Port 1 Registers .....	122
9.3.2	Operations of Port 1 .....	123
9.4	Port 2 .....	125
9.4.1	Port 2 Registers .....	127
9.4.2	Operations of Port 2 .....	128
9.5	Port 3 .....	130
9.5.1	Port 3 Registers .....	132
9.5.2	Operations of Port 3 .....	133
9.6	Port 5 .....	135
9.6.1	Port 5 Registers .....	137
9.6.2	Operations of Port 5 .....	138
9.7	Port 6 .....	140
9.7.1	Port 6 Registers .....	142
9.7.2	Operations of Port 6 .....	143
9.8	Port G .....	145
9.8.1	Port G Registers .....	147
9.8.2	Operations of Port G .....	148
<b>CHAPTER 10</b>	<b>TIME-BASE TIMER .....</b>	<b>151</b>
10.1	Overview of Time-base Timer .....	152

10.2	Configuration of Time-base Timer .....	153
10.3	Registers of the Time-base Timer .....	155
10.3.1	Time-base Timer Control Register (TBTC) .....	156
10.4	Interrupts of Time-base Timer .....	158
10.5	Explanation of Time-base Timer Operations and Setup Procedure Example .....	160
10.6	Notes on Using Time-base Timer .....	163
<b>CHAPTER 11 WATCHDOG TIMER .....</b>		<b>165</b>
11.1	Overview of Watchdog Timer .....	166
11.2	Configuration of Watchdog Timer .....	167
11.3	Register of The Watchdog Timer .....	169
11.3.1	Watchdog Timer Control Register (WDTC) .....	170
11.4	Explanation of Watchdog Timer Operations and Setup Procedure Example .....	172
11.5	Notes on Using Watchdog Timer .....	174
<b>CHAPTER 12 WATCH PRESCALER .....</b>		<b>175</b>
12.1	Overview of Watch Prescaler .....	176
12.2	Configuration of Watch Prescaler .....	177
12.3	Registers of the Watch Prescaler .....	179
12.3.1	Watch Prescaler Control Register (WPCR) .....	180
12.4	Interrupts of Watch Prescaler .....	182
12.5	Explanation of Watch Prescaler Operations and Setup Procedure Example .....	184
12.6	Notes on Using Watch Prescaler .....	186
12.7	Sample Programs for Watch Prescaler .....	187
<b>CHAPTER 13 WATCH COUNTER .....</b>		<b>189</b>
13.1	Overview of Watch Counter .....	190
13.2	Configuration of Watch Counter .....	191
13.3	Registers of Watch Counter .....	193
13.3.1	Watch Counter Data Register (WCDR) .....	194
13.3.2	Watch Counter Control Register (WCSR) .....	195
13.4	Interrupts of Watch Counter .....	197
13.5	Explanation of Watch Counter Operations and Setup Procedure Example .....	198
13.6	Notes on Using Watch Counter .....	200
13.7	Sample Programs for Watch Counter .....	201
<b>CHAPTER 14 WILD REGISTER .....</b>		<b>203</b>
14.1	Overview of Wild Register .....	204
14.2	Configuration of Wild Register .....	205
14.3	Registers of Wild Register .....	207
14.3.1	Wild Register Data Setup Registers(WRDR0 to WRDR2) .....	209
14.3.2	Wild Register Address Setup Registers(WRAR0 to WRAR2) .....	210
14.3.3	Wild Register Address Compare Enable Register (WREN) .....	211
14.3.4	Wild Register Data Test Setup Register (WROR) .....	212
14.4	Operating Description of Wild Register .....	213
14.5	Typical Hardware Connection Example .....	214

<b>CHAPTER 15</b>	<b>8/16-BIT COMPOUND TIMER .....</b>	<b>215</b>
15.1	Overview of 8/16-bit Compound Timer .....	216
15.2	Configuration of 8/16-bit Compound Timer .....	218
15.3	Channels of 8/16-bit Compound Timer .....	221
15.4	Pins of 8/16-bit Compound Timer .....	222
15.5	Registers of 8/16-bit Compound Timer .....	224
15.5.1	8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) .....	225
15.5.2	8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) .....	228
15.5.3	8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) .....	231
15.5.4	8/16-bit Compound Timer 00/01 Data Register ch.0 (T00DR/T01DR) .....	234
15.6	Interrupts of 8/16-bit Compound Timer .....	237
15.7	Operating Description of Interval Timer Function (One-shot Mode) .....	239
15.8	Operating Description of Interval Timer Function (Continuous Mode) .....	241
15.9	Operating Description of Interval Timer Function (Free-run Mode) .....	243
15.10	Operating Description of PWM Timer Function (Fixed-cycle mode) .....	245
15.11	Operating Description of PWM Timer Function (Variable-cycle Mode) .....	247
15.12	Operating Description of PWC Timer Function .....	249
15.13	Operating Description of Input Capture Function .....	251
15.14	Operating Description of Noise Filter .....	253
15.15	States in Each Mode during Operation .....	254
15.16	Notes on Using 8/16-bit Compound Timer .....	256
<b>CHAPTER 16</b>	<b>8/16-BIT PPG .....</b>	<b>257</b>
16.1	Overview of 8/16-bit PPG .....	258
16.2	Configuration of 8/16-bit PPG .....	259
16.3	Channels of 8/16-bit PPG .....	261
16.4	Pins of 8/16-bit PPG .....	262
16.5	Registers of 8/16-bit PPG .....	264
16.5.1	8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
16.5.2	8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
16.5.3	8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00) .....	269
16.5.4	8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00) .....	270
16.5.5	8/16-bit PPG Start Register (PPGS) .....	271
16.5.6	8/16-bit PPG Output Inversion Register (REVC) .....	272
16.6	Interrupts of 8/16-bit PPG .....	273
16.7	Operating Description of 8/16-bit PPG .....	274
16.7.1	8-bit PPG Independent Mode .....	275
16.7.2	8-bit Prescaler + 8-bit PPG Mode .....	277
16.7.3	16-bit PPG Mode .....	279
16.8	Notes on Using 8/16-bit PPG .....	281
16.9	Sample Programs for 8/16-bit PPG Timer .....	282
<b>CHAPTER 17</b>	<b>16-BIT PPG TIMER .....</b>	<b>285</b>
17.1	Overview of 16-bit PPG Timer .....	286
17.2	Configuration of 16-bit PPG Timer .....	287
17.3	Channels of 16-bit PPG Timer .....	289
17.4	Pins of 16-bit PPG Timer .....	290

17.5	Registers of 16-bit PPG Timer .....	291
17.5.1	16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0) .....	292
17.5.2	16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0) .....	293
17.5.3	16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) .....	294
17.5.4	16-bit PPG Status Control Register (Upper, Lower) (PCNTH0, PCNTL0) .....	295
17.6	Interrupts of 16-bit PPG Timer .....	299
17.7	Explanation of 16-bit PPG Timer Operations and Setup Procedure Example .....	300
17.8	Notes on Using 16-bit PPG Timer .....	304
17.9	Sample Programs for 16-bit PPG Timer .....	305
<b>CHAPTER 18</b>	<b>EXTERNAL INTERRUPT CIRCUIT .....</b>	<b>309</b>
18.1	Overview of External Interrupt Circuit .....	310
18.2	Configuration of External Interrupt Circuit .....	311
18.3	Channels of External Interrupt Circuit .....	312
18.4	Pins of External Interrupt Circuit .....	313
18.5	Registers of External Interrupt Circuit .....	314
18.5.1	External Interrupt Control Register (EIC00) .....	315
18.6	Interrupts of External Interrupt Circuit .....	317
18.7	Explanation of External Interrupt Circuit Operations and Setup Procedure Example .....	318
18.8	Notes on Using External Interrupt Circuit .....	320
18.9	Sample Programs for External Interrupt Circuit .....	321
<b>CHAPTER 19</b>	<b>INTERRUPT PIN SELECTION CIRCUIT .....</b>	<b>323</b>
19.1	Overview of Interrupt Pin Selection Circuit .....	324
19.2	Configuration of Interrupt Pin Selection Circuit .....	325
19.3	Pins of Interrupt Pin Selection Circuit .....	326
19.4	Registers of Interrupt Pin Selection Circuit .....	327
19.4.1	Interrupt Pin Selection Circuit Control Register (WICR) .....	328
19.5	Operating Description of Interrupt Pin Selection Circuit .....	331
19.6	Notes on Using Interrupt Pin Selection Circuit .....	332
<b>CHAPTER 20</b>	<b>UART/SIO .....</b>	<b>333</b>
20.1	Overview of UART/SIO .....	334
20.2	Configuration of UART/SIO .....	335
20.3	Channels of UART/SIO .....	337
20.4	Pins of UART/SIO .....	338
20.5	Registers of UART/SIO .....	340
20.5.1	UART/SIO Serial Mode Control Register 1 (SMC10) .....	341
20.5.2	UART/SIO Serial Mode Control Register 2 (SMC20) .....	343
20.5.3	UART/SIO Serial Status and Data Register (SSR0) .....	345
20.5.4	UART/SIO Serial Input Data Register (RDR0) .....	347
20.5.5	UART/SIO Serial Output Data Register (TDR0) .....	348
20.6	Interrupts of UART/SIO .....	349
20.7	Explanation of UART/SIO Operations and Setup Procedure Example .....	350
20.7.1	Operating Description of Operation Mode 0 .....	351
20.7.2	Operating Description of Operation Mode 1 .....	358
20.8	Sample Programs for UART/SIO .....	364

<b>CHAPTER 21</b>	<b>UART/SIO DEDICATED BAUD RATE GENERATOR .....</b>	<b>369</b>
21.1	Overview of UART/SIO Dedicated Baud Rate Generator .....	370
21.2	Channels of UART/SIO Dedicated Baud Rate Generator .....	371
21.3	Registers of UART/SIO Dedicated Baud Rate Generator .....	372
21.3.1	UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0) .....	373
21.3.2	UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0) .....	374
21.4	Operating Description of UART/SIO Dedicated Baud Rate Generator .....	375
<b>CHAPTER 22</b>	<b>LIN-UART .....</b>	<b>377</b>
22.1	Overview of LIN-UART .....	378
22.2	Configuration of LIN-UART .....	380
22.3	Pins of LIN-UART .....	385
22.4	Registers of LIN-UART .....	386
22.4.1	LIN-UART Serial Control Register (SCR) .....	387
22.4.2	LIN-UART Serial Mode Register (SMR) .....	389
22.4.3	LIN-UART Serial Status Register (SSR) .....	391
22.4.4	LIN-UART Reception Data Register/LIN-UART Transmit Data Register (RDR/TDR) .....	393
22.4.5	LIN-UART Extended Status Control Register (ESCR) .....	395
22.4.6	LIN-UART Extended Communication Control Register (ECCR) .....	397
22.4.7	LIN-UART Baud Rate Generator Register 1, 01, 0 (BGR1, BGR0) .....	399
22.5	Interrupt of LIN-UART .....	400
22.5.1	Reception Interrupt Generation and Flag Set Timing .....	404
22.5.2	Transmit Interrupt Generation and Flag Set Timing .....	406
22.6	LIN-UART Baud Rate .....	408
22.6.1	Baud Rate Setting .....	410
22.6.2	Reload Counter .....	414
22.7	Operations and Setup Procedure Example of LIN-UART .....	416
22.7.1	Operation of Asynchronous Mode (Operation Mode 0, 1) .....	418
22.7.2	Operation of Synchronous Mode (Operation Mode 2) .....	422
22.7.3	Operation of LIN function (Operation Mode 3) .....	426
22.7.4	Serial Pin Direct Access .....	429
22.7.5	Bi-directional Communication Function (Normal Mode) .....	430
22.7.6	Master/slave Mode Communication Function (Multi-processor Mode) .....	432
22.7.7	LIN Communication Function .....	435
22.7.8	Example of LIN-UART LIN Communication Flowchart (Operation Mode 3) .....	436
22.8	Notes on Using LIN-UART .....	438
22.9	Sample Programs of LIN-UART .....	443
<b>CHAPTER 23</b>	<b>I<sup>2</sup>C .....</b>	<b>449</b>
23.1	Overview of I <sup>2</sup> C .....	450
23.2	I <sup>2</sup> C Configuration .....	451
23.3	I <sup>2</sup> C Channels .....	454
23.4	I <sup>2</sup> C Bus Interface Pins .....	455
23.5	I <sup>2</sup> C Registers .....	457
23.5.1	I <sup>2</sup> C Bus Control Registers 0, 1 (IBCR00, IBCR10) .....	458
23.5.2	I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
23.5.3	I <sup>2</sup> C Data Register (IDDR0) .....	466

23.5.4	I <sup>2</sup> C Address Register (IAAR0)	467
23.5.5	I <sup>2</sup> C Clock Control Register (ICCR0)	468
23.6	I <sup>2</sup> C Interrupts	470
23.7	I <sup>2</sup> C Operations and Setup Procedure Examples	473
23.7.1	I <sup>2</sup> C Interface	474
23.7.2	Function to Wake up the MCU from Standby Mode	481
23.8	Notes on Use of I <sup>2</sup> C	483
23.9	Sample Programs for I <sup>2</sup> C	485
<b>CHAPTER 24</b>	<b>8/10-BIT A/D CONVERTER</b>	<b>489</b>
24.1	Overview of 8/10-bit A/D Converter	490
24.2	Configuration of 8/10-bit A/D Converter	491
24.3	Pins of 8/10-bit A/D Converter	493
24.4	Registers of 8/10-bit A/D Converter	495
24.4.1	8/10-bit A/D Converter Control Register 1 (ADC1)	496
24.4.2	8/10-bit A/D Converter Control Register 2 (ADC2)	498
24.4.3	8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)	500
24.5	Interrupts of 8/10-bit A/D Converter	501
24.6	Operations of 8/10-bit A/D Converter and Its Setup Procedure Examples	502
24.7	Notes on Use of 8/10-bit A/D Converter	505
24.8	Sample Programs for 8/10-bit A/D Converter	506
<b>CHAPTER 25</b>	<b>LOW-VOLTAGE DETECTION RESET CIRCUIT</b>	<b>509</b>
25.1	Overview of Low-voltage Detection Reset Circuit	510
25.2	Configuration of Low-voltage Detection Reset Circuit	511
25.3	Pins of Low-voltage Detection Reset Circuit	512
25.4	Operations of Low-voltage Detection Reset Circuit	513
<b>CHAPTER 26</b>	<b>CLOCK SUPERVISOR</b>	<b>515</b>
26.1	Overview of Clock Supervisor	516
26.2	Configuration of Clock Supervisor	517
26.3	Register of Clock Supervisor	519
26.3.1	Clock Supervisor Control Register (CSVCR)	520
26.4	Operations of Clock Supervisor	522
26.5	Notes on Using Clock Supervisor	525
<b>CHAPTER 27</b>	<b>DUAL-OPERATION FLASH MEMORY</b>	<b>527</b>
27.1	Overview of Dual-Operation Flash Memory	528
27.2	Sector/Bank Configuration of Flash Memory	530
27.3	Register of Flash Memory	531
27.3.1	Flash Memory Status Register (FSR)	532
27.3.2	Flash Memory Sector Write Control Registers (SWRE0/SWRE1)	535
27.4	Starting the Flash Memory Automatic Algorithm	540
27.5	Checking the Automatic Algorithm Execution Status	542
27.5.1	Data Polling Flag (DQ7)	544
27.5.2	Toggle Bit Flag (DQ6)	546
27.5.3	Execution Time-out Flag (DQ5)	548

27.5.4	Sector Erase Timer Flag (DQ3)	549
27.6	Flash Memory Program/Erase	550
27.6.1	Placing Flash Memory in the Read/Reset State	551
27.6.2	Programming Data into Flash Memory	552
27.6.3	Erasing All Data from Flash Memory (Chip Erase)	554
27.6.4	Erasing Arbitrary Data from Flash Memory (Sector Erasing)	555
27.6.5	Suspending Sector Erasing from Flash Memory	557
27.6.6	Resuming Sector Erasing from Flash Memory	558
27.7	Operation of Dual-Operation Flash Memory	559
27.8	Flash Security	561
27.9	Notes on Using Dual-Operation Flash Memory	562
<b>CHAPTER 28</b>	<b>256-Kbit FLASH MEMORY</b>	<b>563</b>
28.1	Overview of 256-Kbit Flash Memory	564
28.2	Sector Configuration of Flash Memory	565
28.3	Register of Flash Memory	566
28.3.1	Flash Memory Status Register (FSR)	567
28.4	Starting the Flash Memory Automatic Algorithm	569
28.5	Checking the Automatic Algorithm Execution Status	570
28.5.1	Data Polling Flag (DQ7)	571
28.5.2	Toggle Bit Flag (DQ6)	572
28.5.3	Execution Time-out Flag (DQ5)	573
28.6	Flash Memory Program/Erase	574
28.6.1	Placing Flash Memory in the Read/Reset State	575
28.6.2	Programming Data into Flash Memory	576
28.6.3	Erasing All Data from Flash Memory (Chip Erase)	578
28.7	Flash Security	579
<b>CHAPTER 29</b>	<b>EXAMPLE OF SERIAL PROGRAMMING CONNECTION</b>	<b>581</b>
29.1	Basic Configuration of Serial Programming Connection for Flash Memory Products	582
29.2	Example of Serial Programming Connection	585
29.3	Example of Minimum Connection to Flash Microcontroller Programmer	588
<b>CHAPTER 30</b>	<b>DUAL OPERATION FLASH</b>	<b>591</b>
30.1	Overview of Dual Operation Flash	592
30.2	Access Sector Map of Dual Operation Flash	593
30.3	Operations of Dual Operation Flash	595
30.4	Notes on Using Dual Operation Flash	597
<b>APPENDIX</b>		<b>599</b>
APPENDIX A	I/O Map	600
APPENDIX B	Table of Interrupt Causes	605
APPENDIX C	Memory Map	606
APPENDIX D	Pin Status of MB95110B/M series	608
APPENDIX E	Instruction Overview	610
E.1	Addressing	613
E.2	Special Instruction	617

E.3	Bit Manipulation Instructions (SETB, CLRB) .....	621
E.4	F <sup>2</sup> MC-8FX Instructions .....	622
E.5	Instruction Map .....	625
APPENDIX F Mask Option .....		626
APPENDIX G Writing to Flash Microcontroller Using Parallel Writer .....		628
<b>INDEX.....</b>		<b>631</b>
<b>Register Index.....</b>		<b>653</b>
<b>Pin Function Index .....</b>		<b>656</b>
<b>Interrupt Vector Index .....</b>		<b>657</b>





# Main changes in this edition

Page	Changes (For details, refer to main body.)	
20	1.8 I/O Circuit Type	Corrected Remarks of Classification B'. Reset input → Reset output
24	2.1 Device Handling Precautions ■ Device Handling Precautions	Added "● Serial Communication"
27	■ Pin Connection	Added "● C pin"
97	7.2 Reset Source Register (RSRR) ■ Configuration of Reset Source Register (RSRR) Table 7.2-1	• Corrected the following descriptions of bit5 to bit1 Read access to this bit sets it to "0". → Read or write access (0 or 1) to this bit sets it to "0".
		• Corrected description of bit0 Read access to this bit or a power-on reset sets it to "0". → Read or write access (0 or 1) to this bit or a power-on reset sets it to "0".
103	8.1 Interrupts ■ Interrupt Requests from Peripheral Resources Table 8.1-1	Added (Reset vector) and (Mode data) to Interrupt request.
124	9.3.2 Operations of Port 1 ■ Operations of Port 1	Changed "● Operation of the pull-up control register".
129	9.4.2 Operations of Port 2 ■ Operations of Port 2	Changed "● Operation of the pull-up control register".
134	9.5.2 Operations of Port 3 ■ Operations of Port 3	Changed "● Operation of the pull-up control register".
148	9.8.2 Operations of Port G ■ Operations of Port G	Changed "● Operation of the pull-up control register".
252	15.13 Operating Description of Input Capture Function ■ Operation of Input Capture Function	Added the explanation.
256	15.16 Notes on Using 8/16-bit Compound Timer ■ Notes on Using 8/16-bit Compound Timer	Added the explanation.
259	16.2 Configuration of 8/16-bit PPG ■ Block Diagram of 8/16-bit PPG	Changed Figure 16.2-1. (MCLK, PCK0 to PCK6 → n/MCLK, $2^7/F_{CH}$ , $2^8/F_{CH}$ )

Page	Changes (For details, refer to main body.)	
277	16.7.2 8-bit Prescaler + 8-bit PPG Mode ■ Operation of 8-bit Prescaler + 8-bit PPG Mode	Corrected channel number in description. When the PPG timer 00 (ch.1) down counter operation enable bit (PEN01) → When the PPG timer 01 (ch.0) down counter operation enable bit (PEN01)
340	20.5 Registers of UART/SIO ■ Registers Related to UART/SIO Figure 20.5-1	Corrected bit attribute of bit5 in SMC20 R/W → R1/W
		Corrected explanation R1/W : Readable/writable (Read value is always "1")
344	20.5.2 UART/SIO Serial Mode Control Register 2 (SMC20) ■ UART/SIO Serial Mode Control Register 2 (SMC20) Table 20.5-2	Corrected bit description of bit5 Setting the bit to "1" clears the reception error flag. → Setting the bit to "1": has no effect on operation.
388	22.4.1 LIN-UART Serial Control Register (SCR) ■ LIN-UART Serial Control Register (SCR) Table 22.4-1	<ul style="list-style-type: none"> <li>• Deleted Note in Function cell of bit5</li> <li>• Changed Note in Function cell of bit2</li> </ul>
401	22.5 Interrupt of LIN-UART ■ Reception Interrupt ● Reception interrupt	Changed Note:
417	22.7 Operations and Setting Procedure Example of LIN-UART ■ Setup Procedure Example ● Initial setting	Corrected explanation 1) Set the port input (DDR1). → 1) Set the port for input (DDR6).
439 to 442	22.8 Notes on Using LIN-UART ■ Notes on Using LIN-UART	Added ● Handling framing errors Added Figure22.8-1 to Figure 22.8-3
443	22.9 Sample Programs of LIN-UART ■ Setting Methods not Covered by Sample Programs ● How to control the SCK, SIN, and SOT pins	Corrected table contents DDR6:P05 = 0 → DDR6:P65 = 0 DDR6:P07 = 0 → DDR6:P67 = 0
459	23.5.1 I <sup>2</sup> C Bus Control Registers 0, 1 (IBCR00, IBCR10) ■ I <sup>2</sup> C Bus Control Register 0 (IBCR00)	Changed the function explanation of bit7 in the Table 23.5-1. Write "1" to this bit in either of the following ways: → Update this bit in either of the following ways:
460		Changed the note comment under Table 23.5-1. IBSR → IBCR10
463	■ I <sup>2</sup> C Bus Control Register 1 (IBCR10)	Changed the note comment under Table 23.5-2. IBSR0 → IBCR10

Page	Changes (For details, refer to main body.)	
504	24.6 Operations of 8/10-bit A/D Converter and Its Setup Procedure Examples ■ Setup Procedure Example ● Initial setting	Corrected explanation 1) Set the port for input (DDR1) → 1) Set the port for input (DDR3)
527	CHAPTER 27 DUAL-OPERATION FLASH MEMORY	Changed the chapter name 480-KBIT FLASH MEMORY → DUAL-OPERATION FLASH MEMORY
528	27.1 Overview of Dual-Operation Flash Memory	Changed the summary.
528	■ Overview of Dual-Operation Flash Memory	Changed the summary.
529	■ Features of Dual-Operation Flash Memory	Added the following descriptions • Compatible with JEDEC standard commands
529	■ Flash memory program/erase	Added description • By using dual-operation flash memory, programs can be executed on the flash memory and programming control using interrupt is enabled. In addition, programs do not need to be downloaded onto the RAM for execution when programming, and
542	27.5 Checking the Automatic Algorithm Execution Status ■ Hardware Sequence Flag ● Overview of hardware sequence flag	Changed explanation the following 5-bit outputs: → the following 4-bit outputs:
		Deleted "• Toggle bit 2 flag (DQ2)" DQ7, DQ6, DQ5, DQ3, DQ2 → DQ7, DQ6, DQ5, DQ3
542	Table 27.5-1	Changed bit 2 DQ2 → -
543	Table 27.5-2	Deleted column of "DQ2" Deleted Note(*) at the bottom of the table
-	27.5.5 Toggle Bit 2 Flag (DQ2)	Deleted whole section of 27.5.5
553	Figure 27.6-1	Changed flow chart Changed hexadecimal number Added "H"
556	Figure 27.6-2	Changed flow chart
559, 560	27.7 Operation of Dual-Operation Flash Memory	Added section
562	27.9 Notes on Using Dual-Operation Flash Memory	Added explanation ■ Notes on F <sup>2</sup> MC-8FX Software Development Support Environment (MB95FV100D and MB2146-09) • Writing or erasing the lower bank (1000H to 3FFFH) is not possible. • Do not execute chip erasing.

Page	Changes (For details, refer to main body.)	
564	CHAPTER 28 256-KBIT FLASH MEMORY 28.1 Overview of 256-Kbit Flash Memory	Changed summary
564	■ Overview of 256-Kbit Flash Memory	Changed descriptions
568	28.3.1 Flash Memory Status Register (FSR) ■ Flash Memory Status Register (FSR) Table 25.3-1	Deleted the following description of bit1 • To program data into the flash memory, set FSR:WRE to "1" to write-enable the flash memory and set the flash memory sector write control register (SWRE0/SWRE1). When FSR:WRE disables programming (contains "0"),
569	28.4 Starting the Flash Memory Automatic Algorithm ■ Command Sequence Table Table 28.4-1	Changed explanation of "U" U : Upper 4 bits same as RA, PA, and SA → U : Upper 4 bits same as RA and PA
569	28.4 Starting the Flash Memory Automatic Algorithm ■ Command Sequence Table	Corrected descriptions of Notes
570	28.5 Checking the Automatic Algorithm Execution Status ■ Hardware Sequence Flag ● Overview of hardware sequence flag	Changed description consists of the following 4-bit outputs:→ consists of the following 3-bit outputs: Deleted "Toggle bit 2 flag (DQ2)" Deleted "Note, however, that hardware sequence flags are output only for the bank on a command-issued side." DQ7, DQ6,
570	Table 28.5-1	Changed bit 2 DQ2 → -
570	Table 28.5-2	Deleted column of "DQ2" Deleted Note(*) at the bottom of the table
-	28.5.4 Toggle Bit 2 Flag (DQ2)	Deleted whole section of 28.5.4
605	APPENDIX B Table of Interrupt Causes ■ Table of Interrupt Causes	Added the rows of "Reset vector" and "Mode data" in Table B-1.
		Added the interrupt causes in Table B-1. 8/16-bit PPG ch.0 (upper) → 8/16-bit PPG ch.0 (lower) 8/16-bit PPG ch.0 (lower) → 8/16-bit PPG ch.0 (upper)

The vertical lines marked in the left side of the page show the changes.

# **CHAPTER 1**

---

# ***DESCRIPTION***

**This chapter explains a feature and a basic specification of the MB95110B/M series.**

- 1.1 Feature of MB95110B/M Series
- 1.2 Product Lineup of MB95110B/M Series
- 1.3 Difference Points among Products and Notes on Selecting a Product
- 1.4 Block Diagram of MB95110B/M Series
- 1.5 Pin Assignment
- 1.6 Package Dimension
- 1.7 Pin Description
- 1.8 I/O Circuit Type

## **1.1 Feature of MB95110B/M Series**

---

**In addition to a compact instruction set, the MB95110B/M series is a general-purpose single-chip microcontroller built-in abundant peripheral functions.**

---

### **■ Feature of MB95110B/M Series**

#### ● F<sup>2</sup>MC-8FX CPU core

Instruction system optimized for controllers

- Multiplication and division instructions
- 16-bit operation
- Bit test branch instruction
- Bit operation instructions etc.

#### ● Clock

- Main clock
- Main PLL clock
- Sub clock (Only for dual clock product)
- Sub PLL clock (Only for dual clock product, except MB95F116MAW/F116NAW)

#### ● Timer

- 8/16-bit compound timer × 2 channels
- 8/16-bit PPG × 2 channels
- 16-bit PPG
- Time-base timer
- Watch prescaler (Only for dual clock product)

#### ● LIN-UART

- With full-duplex double buffer
- An asynchronous clock or a synchronous serial data transfer can be used

#### ● UART/SIO

- With full-duplex double buffer
- An asynchronous clock or a synchronous serial data transfer can be used

#### ● I<sup>2</sup>C

Built-in wake up function

#### ● External interrupt

- Interrupt by the edge detection (Select from rising edge, falling edge, or both edges)
- Can be used to recover from low-power consumption (standby) mode

- 8/10-bit A/D converter

8-bit or 10-bit resolutions can be selected

- Low-power consumption (standby) mode

- Stop mode
- Sleep mode
- Watch mode (Only for dual clock product)
- Time-base timer mode

- I/O port

- Maximum number of ports
  - single clock product : 39
  - Dual clock product : 37
- General-purpose I/O ports (N-ch open drain): 2
- General-purpose I/O ports (CMOS)
  - single clock product : 37
  - Dual clock product : 35

- Programmable input voltage levels of port

Automotive input level / CMOS input level / hysteresis input level

- Flash memory security function

Protects the content of Flash memory (Flash memory device only)



## 1.2 Product Lineup of MB95110B/M Series

MB95110B/M series is available in three types. Table 1.2-1 lists the product lineup and Table 1.2-2 lists the CPUs and peripheral functions.

### ■ Product Lineup of MB95110B/M Series

**Table 1.2-1 Product Lineup of MB95110B/M Series**

Classification		Product	ROM/RAM	Voltage	Option			Reset output	
					Clock system	LVD	CSV		
Evaluation products *1		MB95FV100D-101	60Kbytes/ 3.75Kbytes	3V	Single-system	None	None	None	
				Dual-system					
		MB95FV100D-103	60Kbytes/ 3.75Kbytes	5V	Single-system	None	None	Yes	
						Yes	None		
						Yes	Yes		
					Dual-system	None	None		
						Yes	None		
		Yes	Yes						
3V products	Flash memory products	MB95F118BS	60Kbytes/2Kbytes	3V	Single-system	None	None	None	
		MB95F118BW			Dual-system				
	Mask ROM products *2	MB95116B	32Kbytes/1Kbytes		Single-system	None	None	None	
					Dual-system				
5V products	Flash memory products	MB95F114MS	16Kbytes/512bytes	5V	Single-system	None	None	Yes	
		MB95F114NS				Yes	None	Yes	
		MB95F114JS				Yes	Yes	Yes	
		MB95F114MW			Dual-system	None	None	Yes	
		MB95F114NW				Yes	None	Yes	
		MB95F114JW				Yes	Yes	Yes	
		MB95F116MS	32Kbytes/1Kbytes		Single-system	None	None	Yes	
		MB95F116NS				Yes	None	Yes	
		MB95F116JS				Yes	Yes	Yes	
		MB95F116MW			Dual-system	None	None	Yes	
		MB95F116NW				Yes	None	Yes	
		MB95F116JW				Yes	Yes	Yes	
		MB95F118MS	60Kbytes/2Kbytest		Single-system	None	None	Yes	
		MB95F118NS				Yes	None	Yes	
		MB95F118JS				Yes	Yes	Yes	
		MB95F118MW			Dual-system	None	None	Yes	
		MB95F118NW				Yes	None	Yes	
		MB95F118JW				Yes	Yes	Yes	
		MB95F116MAS	32Kbytes/1Kbytes		Single-system	None	None	Yes	
		MB95F116NAS				Yes	None	Yes	
		MB95F116MAW			Dual-system	None	None	Yes	
		MB95F116NAW				Yes	None	Yes	
	Mask ROM products *2	MB95117M	48Kbytes/2Kbytes		Single-system	None	None	Yes	
						Yes	None		
						Yes	Yes	None	
					Dual-system	None	None	Yes	
						Yes	None		
						Yes	Yes	None	

LVD: Low-voltage detection reset

CSV: Clock Supervisor

- \*1: For evaluation products, use the switch on MCU board to enable/disable LVD, CSV, and the 1/2 system (LVD cannot be disabled while CSV is enabled).
- \*2: For the mask ROM products, enable/disable LVD, CSV, and the 1/2 system when ordering the mask ROM (LVD cannot be disabled while CSV is enabled).

### Table 1.2-2 CPU and Peripheral Function of MB95110B/M Series (1 / 2)

Parameter		Specification	
CPU function		Number of basic instructions: 136 instructions Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, and 16 bits Minimum instruction execution time: 61.5 ns (at machine clock 16.25 MHz) Interrupt processing time: 0.6 μs (at machine clock 16.25 MHz)	
Peripheral function	Port	General-purpose I/O ports (N-ch open drain): 2 General-purpose I/O ports (CMOS) : single clock product: 37 : dual clock product: 35 Total : single clock product: 39 : dual clock product: 37	
	Time-base timer	Interrupt cycle: 0.5 ms, 2.1 ms, 8.2 ms, 32.8 ms (at external 4 MHz)	
	Watchdog timer	Reset generation cycle Main clock at 10 MHz : 105 ms (Min) Sub clock at 32.768 kHz (Only for dual clock product) : 250 ms (Min)	
	Wild registers	ROM data for three bytes can be replaced	
	I <sup>2</sup> C bus	Master/slave sending/receiving Bus error function, Arbitration function, Forwarding direction detection function Generating repeatedly and detecting function of the start condition Built-in wake up function	
	UART/SIO	Data transfer is enabled at UART/SIO Built-in full-duplex double buffer, Changeable data length (5/6/7/8-bit), Built-in baud rate generator NRZ method transfer format, Error detected function LSB-first or MSB-first can be selected Serial data transfer is available for clock synchronous (SIO) and clock asynchronous (UART)	
	LIN-UART	A wide-range communication speed can be set with the dedicated reload timer Full-duplex double buffer Serial data transfer is available for clock synchronous and clock asynchronous LIN function can be used as a LIN master and LIN slave	
	8/10-bit A/D converter	8ch. 8-bit or 10-bit resolution can be selected	
	8/16-bit compound timer	2ch. Can be configured as a 2ch × 8-bit timer or 1ch × 16-bit timer per each timer channel Built-in timer function, PWC function, PWM function and capture function Count clock: available from internal clocks (7 types) or external clocks With square wave output	
	16-bit PPG	PWM mode or one-shot mode can be selected Counter operation clock: available from eight selectable clock sources Support for external trigger activation	
	8/16-bit PPG	2ch. Can be configured as a 2ch × 8-bit PPG or 1ch × 16-bit PPG per each PPG channel Counter operation clock: available from eight selectable clock sources	
	Watch counter	Count clock: available from four selectable clock sources (125 ms, 250 ms, 500 ms, or 1 s) Counter value can be set within the range of 0 to 63 (When one second is selected as for the clock source and the counter value is set to 60, it is possible to count for one minute. ) Note: At selecting the dual clock product	
	Watch prescaler	Available from four selectable interval times (125 ms, 250 ms, 500 ms, 1 s) Note: At selecting the dual clock product	
External interrupt	8ch. Interrupt by edge detection (Possible to select from rising edge, falling edge or both edges) Can be used to recover from standby mode		

**Table 1.2-2 CPU and Peripheral Function of MB95110B/M Series (2 / 2)**

Parameter		Specification
Peripheral function	Flash memory	Supports automatic programming, Embedded Algorithm Write/Erase/Erase-Suspend/Resume commands A flag indicating completion of the algorithm Number of write/erase cycles : 10000 times Data retention time : 20 years Erase can be performed on each block Flash Memory Security feature
Standby Mode		Sleep, stop, watch (Only for dual clock product), and time-base timer

## **1.3 Difference Points among Products and Notes on Selecting a Product**

---

**The following describes differences among MB95110B/M series products and notes when selecting the product.**

---

### **■ Difference Points among Products and Notes on Selecting a Product**

#### **● Notes on using evaluation products**

The evaluation products are intended to support software development for a number of different F<sup>2</sup>MC-8FX family series and products, and it therefore includes additional functions that may not be included in MB95110B/M series. Accordingly, access to I/O address of peripheral functions that are not used in MB95110B/M series are prohibited.

Reading or writing to these prohibited addresses may cause these unused peripheral functions to operate and lead to unexpected hardware or software problems.

Take particular care not to use word access to read or write odd numbered bytes in the prohibited areas (It causes unexpected read/write operation). Also, as the read values of prohibited addresses on the evaluation product are different to the values on the flash memory and mask ROM products, do not use these values in the program.

The functions corresponding to certain bits in single-byte registers may not be supported on some mask ROM and flash memory products. However, reading or writing to these bits will not cause malfunction of the hardware. Also, as the evaluation, flash memory, and mask ROM products are designed to have identical software operation, no particular precautions are required.

#### **● Difference of memory space**

If the memory size on the evaluation product is different to the flash memory or mask ROM product, please ensure you understand these differences when developing software.

#### **● Current consumption**

The current consumption of flash memory products is greater than for mask ROM products.

For the details of current consumption, refer to "ELECTORICAL CHARACTERISTICS" in data sheet.

#### **● Package**

For detailed information on each package, see "■ Package and Its Corresponding Product" and "1.6 Package Dimension".

#### **● Operating voltage**

The operating voltage may be different depending on the products. For the details, see the "Electric characteristics" of "data sheet".

#### ● Difference of $\overline{\text{RST}}$ /MOD pins

For mask ROM products, the  $\overline{\text{RST}}$  and MOD pins are hysteresis inputs (However, on 5 V products, these pins are hysteresis inputs for both mask ROM products and flash memory products). And, a pull-down resistor is provided for the MOD pin.

#### ■ Package and Its Corresponding Product

<div>Product</div> <div>Package</div>	MB95116B	MB95F118BS MB95F118BW	MB95117M	MB95F114MS MB95F114NS MB95F114JS MB95F114MW MB95F114NW MB95F114JW MB95F116MS MB95F116NS MB95F116JS MB95F116MW MB95F116NW MB95F116JW MB95F118MS MB95F118NS MB95F118JS MB95F118MW MB95F118NW MB95F118JW MB95F116MAS MB95F116NAS MB95F116MAW MB95F116NAW	MB95FV100D-101 MB95FV100D-103
FPT-52P-M01	○	○	○	○	×
FPT-48P-M26	○	○	×	×	×
LCC-48P-M09	○	○	×	×	×
BGA-224P-M08	×	×	×	×	○

○: usable

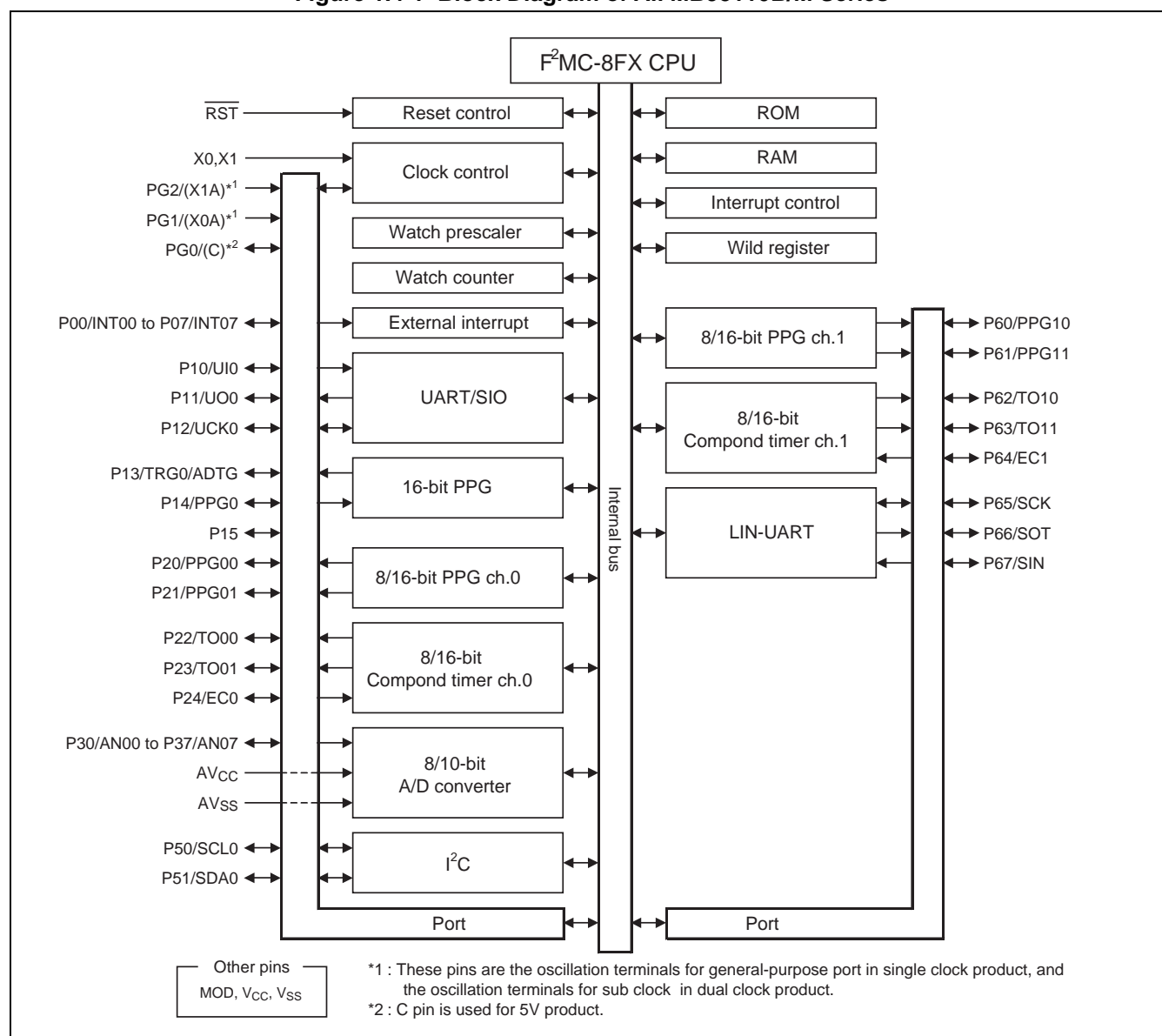
× : unusable

## 1.4 Block Diagram of MB95110B/M Series

Figure 1.4-1 shows the block diagram of all MB95110B/M Series.

### ■ Block Diagram of All MB95110B/M Series

Figure 1.4-1 Block Diagram of All MB95110B/M Series



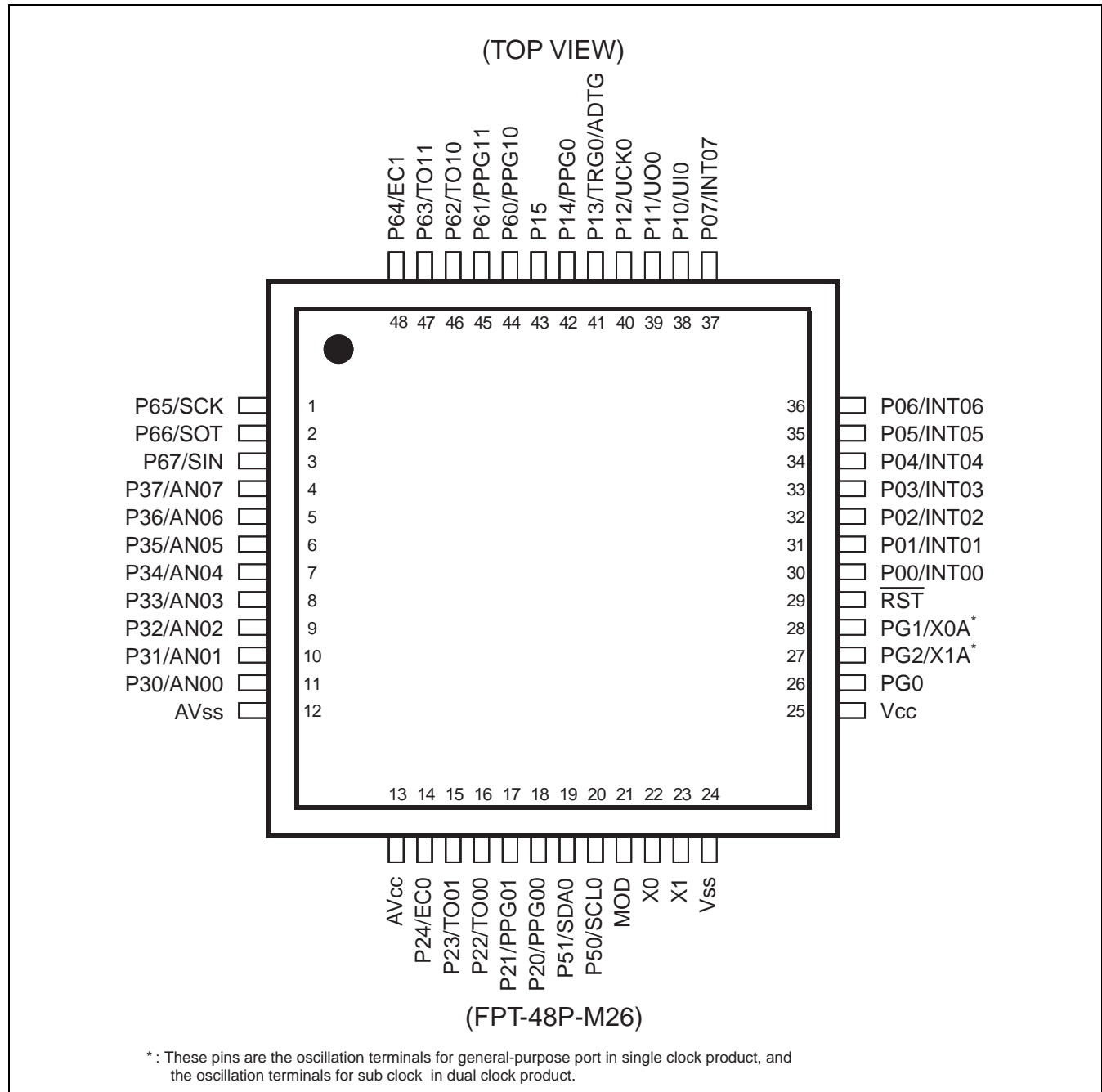
## MB95110B/M Series

### 1.5 Pin Assignment

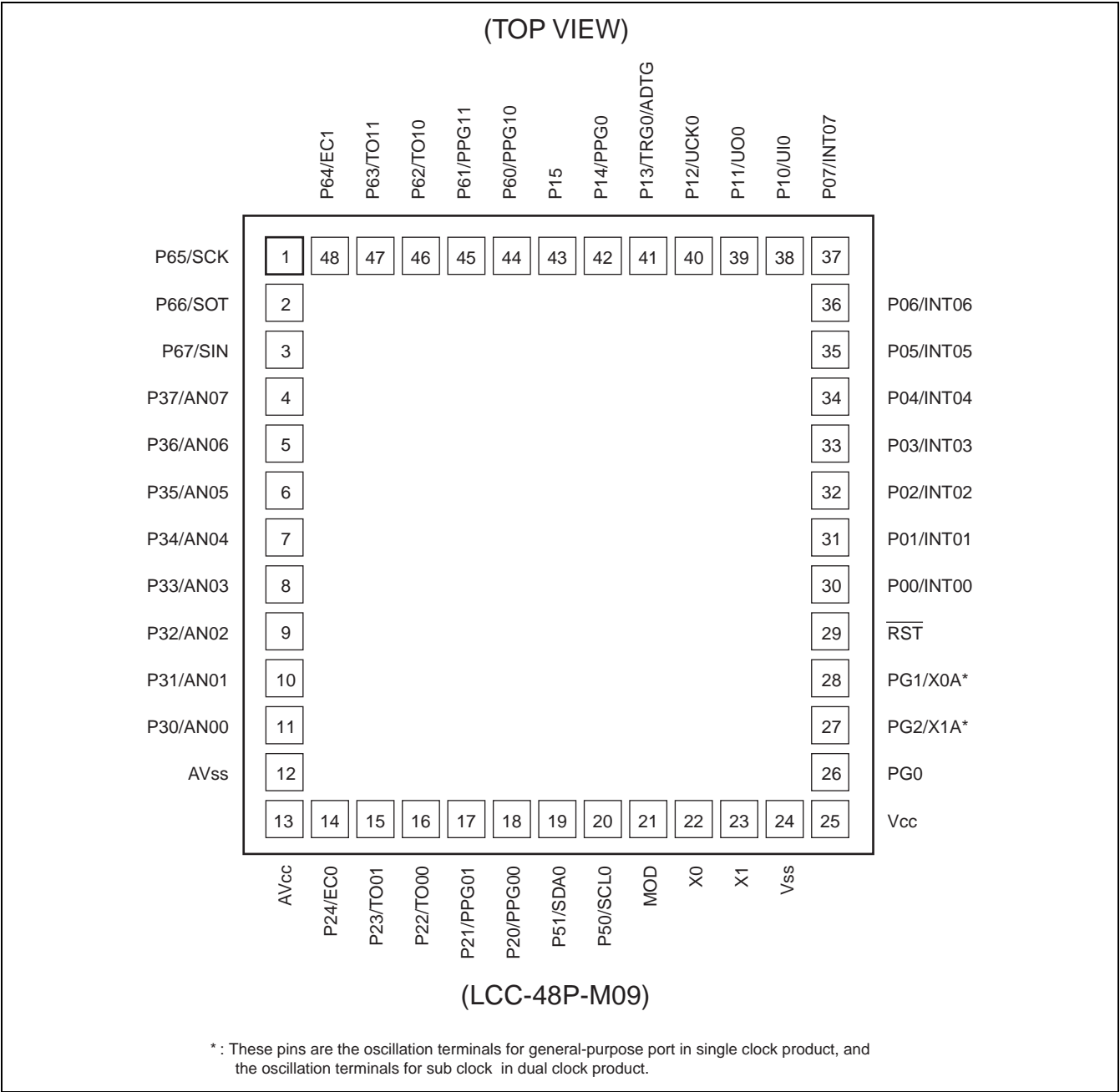
Figure 1.5-1 shows the pin assignment of the MB95110B/M Series.

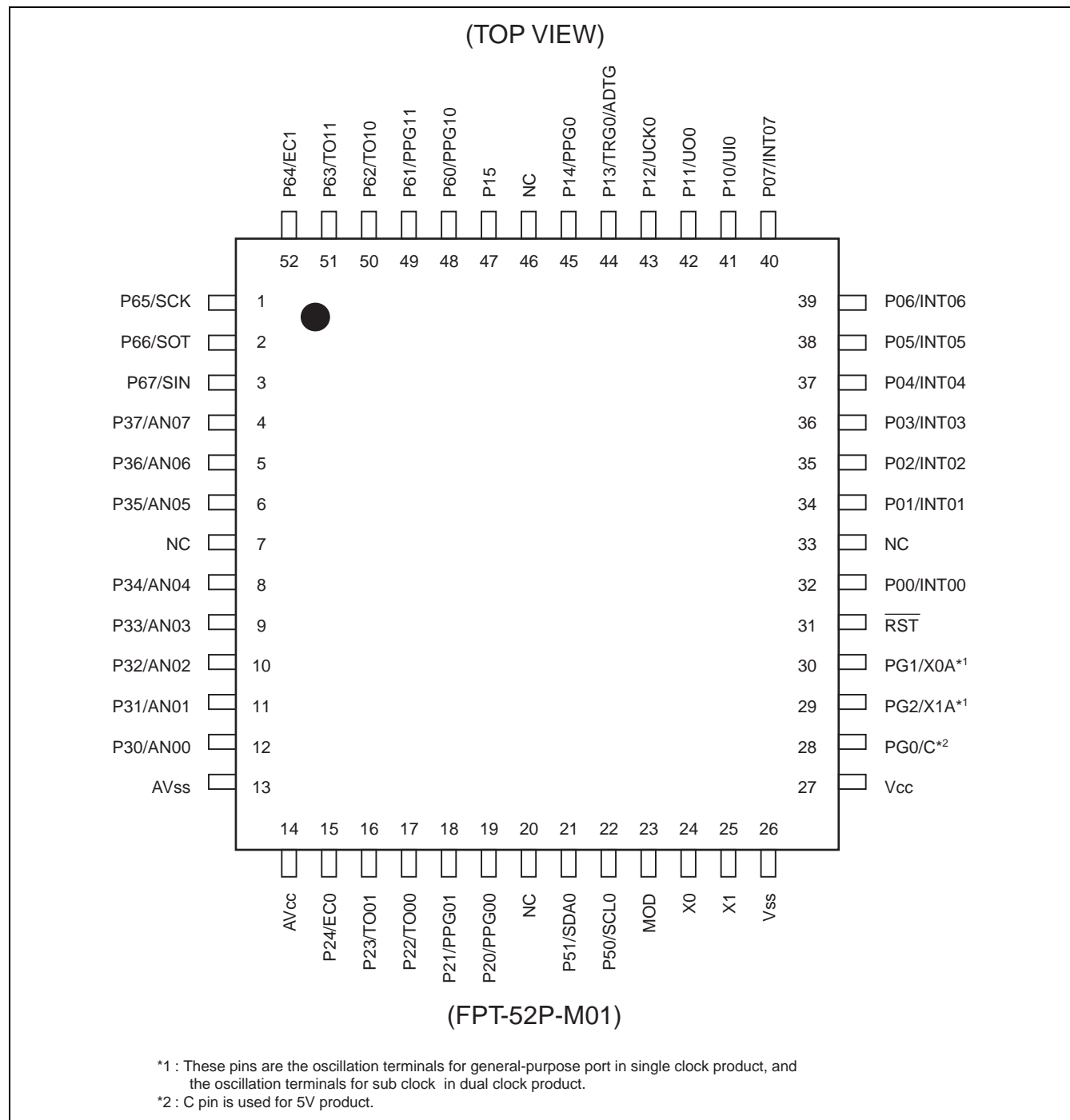
#### ■ Pin Assignment of MB95110B/M Series

Figure 1.5-1 Pin Assignment of MB95110B/M Series







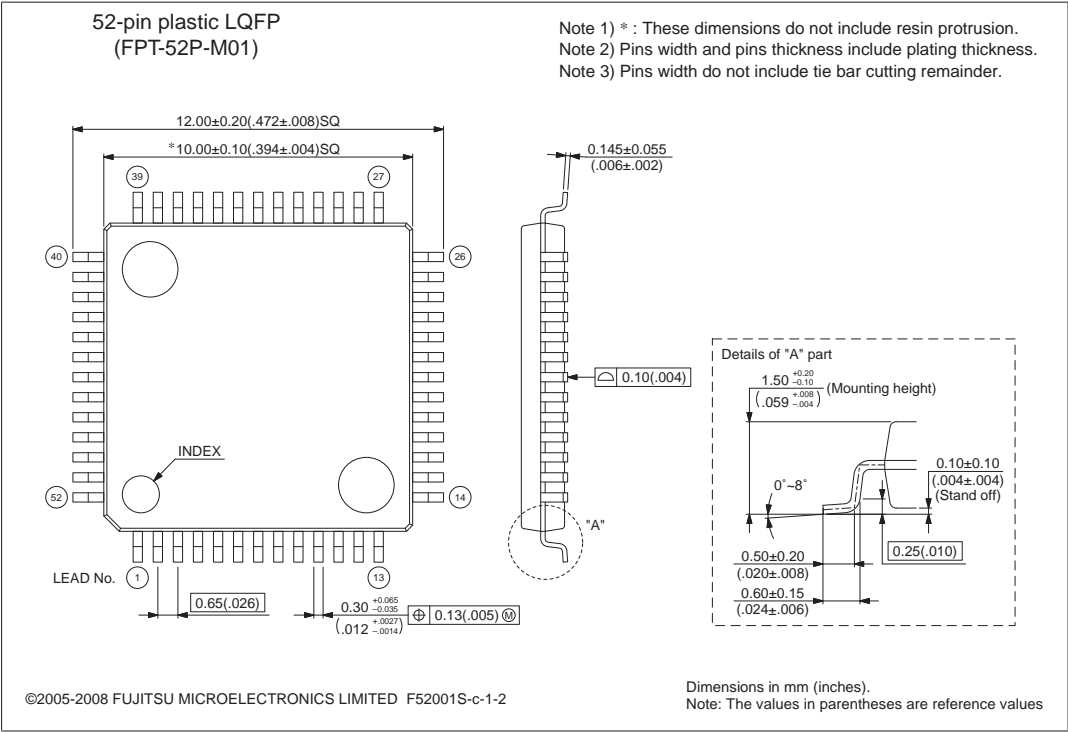
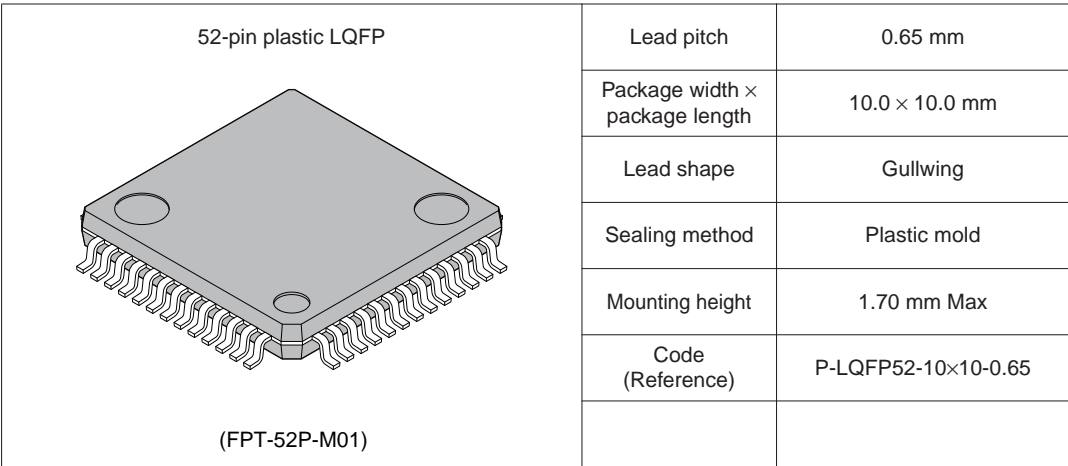


1.6 Package Dimension

MB95110B/M series is available in 3 types of package.

■ Package Dimension of FPT-52P-M01

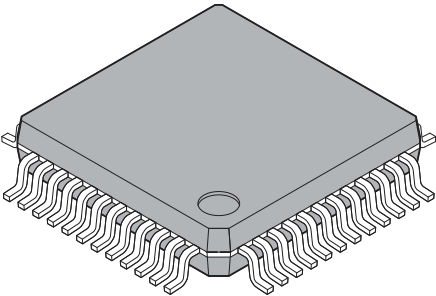
Figure 1.6-1 Package Dimension of FPT-52P-M01

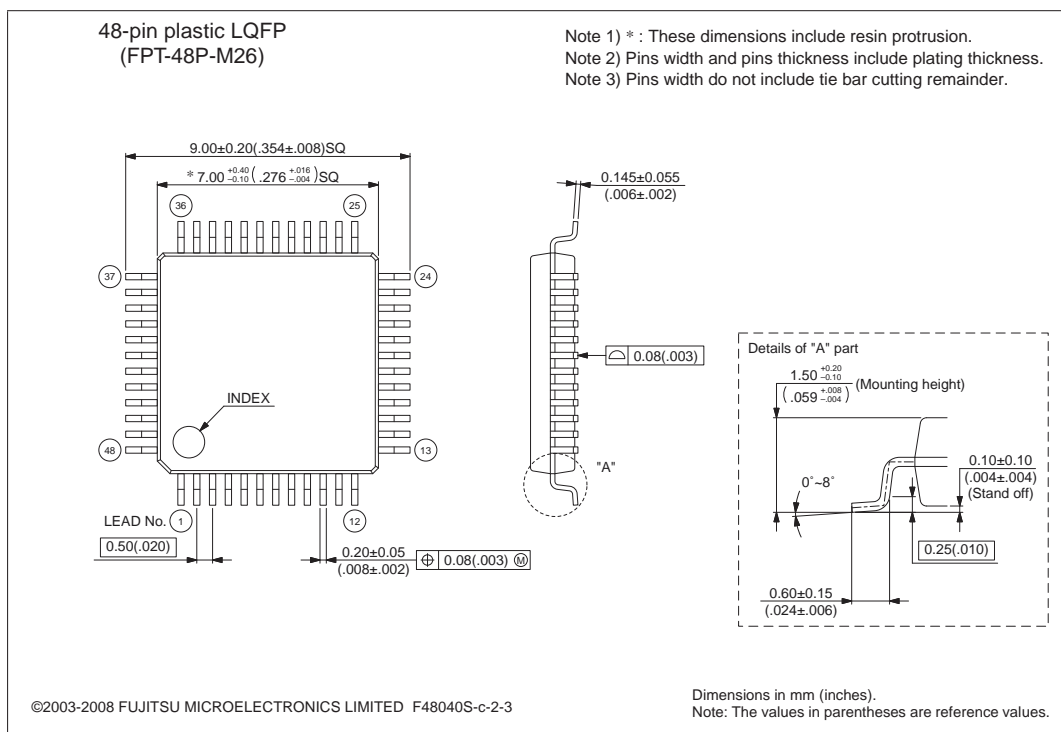


Please confirm the latest Package dimension by following URL.  
<http://edevic.fujitsu.com/package/en-search/>

## ■ Package Dimension of FPT-48P-M26

Figure 1.6-2 Package Dimension of FPT-48P-M26

 <p>48-pin plastic LQFP</p> <p>(FPT-48P-M26)</p>	Lead pitch	0.50 mm
	Package width × package length	7 × 7 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.17 g
	Code (Reference)	P-LFQFP48-7×7-0.50

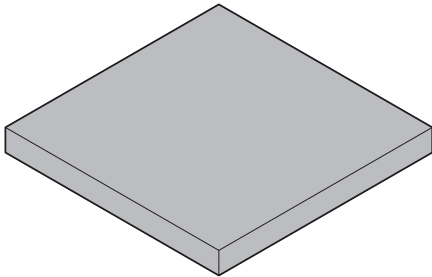


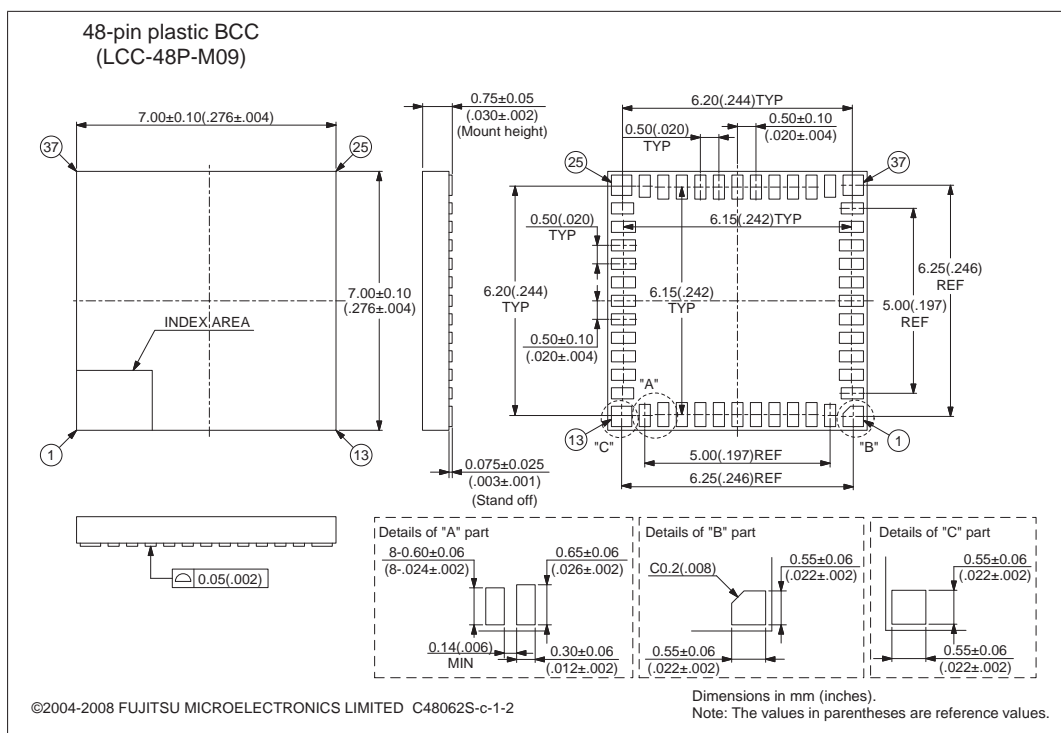
Please confirm the latest Package dimension by following URL.

<http://edevice.fujitsu.com/package/en-search/>

# **■ Package Dimension of LCC-48P-M09**

Figure 1.6-3 Package Dimension of LCC-48P-M09

<div style="text-align: center;"> 48-pin plastic BCC    (LCC-48P-M09) </div>	Lead pitch	0.50 mm
	Package width × package length	7.00 mm × 7.00 mm
	Sealing method	Plastic mold
	Mounting height	0.80 mm Max
	Weight	0.06 g



Please confirm the latest Package dimension by following URL.

<http://edevice.fujitsu.com/package/en-search/>

## 1.7 Pin Description

Table 1.7-1 shows pin description. The alphabet in the "Circuit Type" column of Table 1.7-1 corresponds to the one in the "Classification" column of Table 1.8-1.

### ■ Pin Description

Table 1.7-1 Pin Description (1 / 3)

Pin No.		Pin Name	I/O circuit Type <sup>*3</sup>	Function Description
LQFP <sup>*1</sup> BCC <sup>*1</sup>	LQFP <sup>*2</sup>			
1	1	P65/SCK	K	General-purpose I/O port. This pin is also used for LIN-UART clock I/O.
2	2	P66/SOT		General-purpose I/O port. This pin is also used for LIN-UART data output.
3	3	P67/SIN	L	General-purpose I/O port. This pin is also used for LIN-UART data input.
4	4	P37/AN07	J	General-purpose I/O port. These pins are also used for A/D convertor analog input.
5	5	P36/AN06		
6	6	P35/AN05		
7	8	P34/AN04		
8	9	P33/AN03		
9	10	P32/AN02		
10	11	P31/AN01		
11	12	P30/AN00		
12	13	AVss	—	Power supply (GND) pin for A/D convertor.
13	14	AVcc	—	Power supply pin for A/D convertor.
14	15	P24/EC0	H	General-purpose I/O port. This pin is also used for 8/16-bit compound timer ch.0 clock input.
15	16	P23/TO01		General-purpose I/O port. This pin is also used for 8/16-bit compound timer ch.0 output.
16	17	P22/TO00		
17	18	P21/PPG01		General-purpose I/O port. These pins are also used for 8/16-bit PPG ch.0 output.
18	19	P20/PPG00		

Table 1.7-1 Pin Description (2 / 3)

Pin No.		Pin Name	I/O circuit Type <sup>*3</sup>	Function Description
LQFP <sup>*1</sup> BCC <sup>*1</sup>	LQFP <sup>*2</sup>			
19	21	P51/SDA0	I	General-purpose I/O port. This pin is also used for I <sup>2</sup> C ch.0 data I/O.
20	22	P50/SCL0		General-purpose I/O port. This pin is also used for I <sup>2</sup> C ch.0 clock I/O.
21	23	MOD	B	Operation mode specification pin.
22	24	X0	A	Input oscillation pin for the main clock.
23	25	X1		I/O oscillation pin for the main clock.
24	26	Vss	—	Power supply (GND) pin.
25	27	Vcc	—	Power supply pin.
26	28	PG0/C	H	General-purpose I/O port (at 3 V product). Capacitance connection pin (at 5 V product).
27	29	PG2/X1A	H/A	Single clock product is general-purpose port (PG2). Dual clock product is I/O oscillation pin for the sub clock (32kHz).
28	30	PG1/X0A		Single clock product is general-purpose port (PG1). Dual clock product is input oscillation pin for the sub clock (32kHz).
29	31	RST	B'	Reset pin.
30	32	P00/INT00	C	General-purpose I/O port. These pins are also used for external interrupt input. The large current port.
31	34	P01/INT01		
32	35	P02/INT02		
33	36	P03/INT03		
34	37	P04/INT04		
35	38	P05/INT05		
36	39	P06/INT06		
37	40	P07/INT07		
38	41	P10/UI0	G	General-purpose I/O port. This pin is also used for UART/SIO ch.0 data input.

Table 1.7-1 Pin Description (3 / 3)

Pin No.		Pin Name	I/O circuit Type <sup>*3</sup>	Function Description
LQFP <sup>*1</sup> BCC <sup>*1</sup>	LQFP <sup>*2</sup>			
39	42	P11/UO0	H	General-purpose I/O port. This pin is also used for UART/SIO ch.0 data output.
40	43	P12/UCK0		General-purpose I/O port. This pin is also used for UART/SIO ch.0 clock I/O.
41	44	P13/TRG0/ ADTG		General-purpose I/O port. This pin is also used for 16-bit PPG ch.0 trigger input (TRG0) and A/D trigger input (ADTG).
42	45	P14/PPG0		General-purpose I/O port. This pin is also used for 16-bit PPG ch.0 output.
43	47	P15		General-purpose I/O port.
44	48	P60/PPG10	K	General-purpose I/O port. These pins are also used for 8/16-bit PPG ch.1 output.
45	49	P61/PPG11		General-purpose I/O port. These pins are also used for 8/16-bit compound timer ch.1 output.
46	50	P62/TO10		
47	51	P63/TO11		General-purpose I/O port. This pin is also used for 8/16-bit compound timer ch.1 clock input.
48	52	P64/EC1		
—	7, 20, 33, 46	NC	—	Internal connected pins. Make the pins to Open.

\*1: FPT-48P-M26

\*2: FPT-52P-M01

\*3: For the I/O circuit type, refer to "■I/O Circuit Type".



1.8 I/O Circuit Type

Table 1.8-1 lists the I/O circuit types. Also, the alphabet in the "Classification" column of Table 1.8-1 corresponds to the one in the "Circuit Type" column of Table 1.7-1.

I/O Circuit Type

Table 1.8-1 I/O Circuit Type (1 / 3)

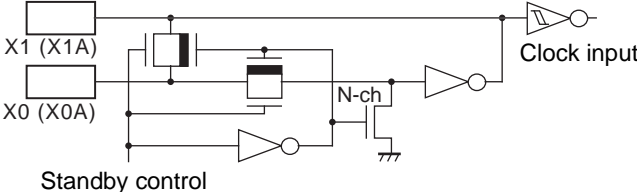
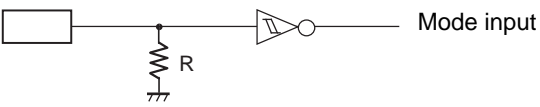
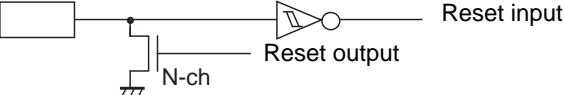
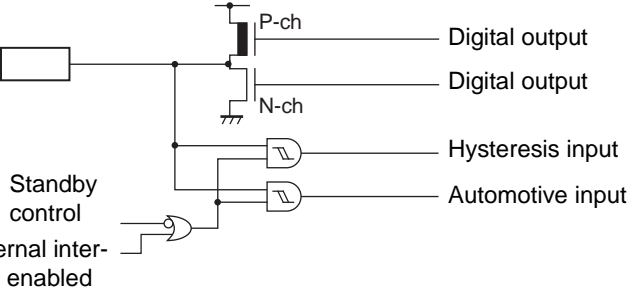
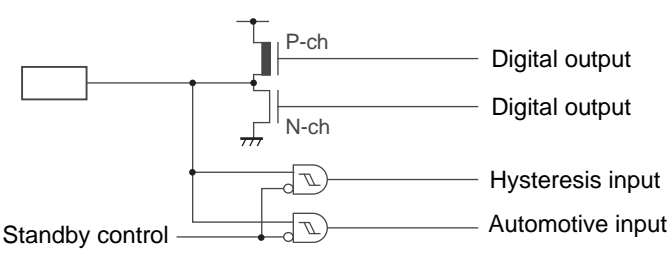
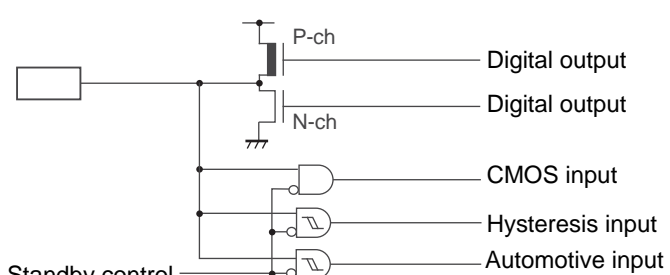
Classi- fication	Circuit	Remarks
A		<ul style="list-style-type: none"><li>• Oscillation circuit</li><li>• High-speed side Feedback resistor: approx 1 MΩ</li><li>• Low-speed side Feedback resistor: approx 24 MΩ (evaluation product: approx 10 MΩ) Dumping resistor : approx 144 kΩ (evaluation product: no dumping resistor)</li></ul>
B		<ul style="list-style-type: none"><li>• Input exclusive use</li><li>• Hysteresis input</li><li>• Pull-down resistance supported (only mask product)</li></ul>
B'		<ul style="list-style-type: none"><li>• Hysteresis input</li><li>• Reset output</li></ul>
C		<ul style="list-style-type: none"><li>• CMOS output</li><li>• Hysteresis input</li><li>• Automotive input</li></ul>

Table 1.8-1 I/O Circuit Type (2 / 3)

Classi- fication	Circuit	Remarks
G		<ul style="list-style-type: none"> <li>• CMOS output</li> <li>• CMOS input</li> <li>• Hysteresis input</li> <li>• Pull-up control supported</li> <li>• Automotive input</li> </ul>
H		<ul style="list-style-type: none"> <li>• CMOS output</li> <li>• Hysteresis input</li> <li>• Pull-up control supported</li> <li>• Automotive input</li> </ul>
I		<ul style="list-style-type: none"> <li>• N-ch open drain output</li> <li>• CMOS input</li> <li>• Hysteresis input</li> <li>• Automotive input</li> </ul>
J		<ul style="list-style-type: none"> <li>• CMOS output</li> <li>• Hysteresis input</li> <li>• Analog input</li> <li>• Pull-up control supported</li> <li>• Automotive input</li> </ul>

Table 1.8-1 I/O Circuit Type (3 / 3)

Classi- fication	Circuit	Remarks
K	 <p>P-ch Digital output</p> <p>N-ch Digital output</p> <p>Standby control</p> <p>Hysteresis input</p> <p>Automotive input</p>	<ul style="list-style-type: none"><li>• CMOS output</li><li>• Hysteresis input</li><li>• Automotive input</li></ul>
L	 <p>P-ch Digital output</p> <p>N-ch Digital output</p> <p>Standby control</p> <p>CMOS input</p> <p>Hysteresis input</p> <p>Automotive input</p>	<ul style="list-style-type: none"><li>• CMOS output</li><li>• CMOS input</li><li>• Hysteresis input</li><li>• Automotive input</li></ul>

# ***CHAPTER 2***

---

# ***HANDLING DEVICES***

**This chapter gives notes on using this series.**

## **2.1 Device Handling Precautions**

## 2.1 Device Handling Precautions

---

**This section describes the precautions common to all devices including the device's power supply voltage and pin treatment.**

**Note that available functions differ depending on the series.**

---

### ■ Device Handling Precautions

#### ● Preventing Latch-up

Care must be taken to ensure that maximum voltage ratings are not exceeded when they are used.

Latch-up may occur on CMOS ICs if voltage higher than  $V_{cc}$  or lower than  $V_{ss}$  is applied to input and output pins other than medium- and high-withstand voltage pins or if higher than the rating voltage is applied between  $V_{cc}$  pin and  $V_{ss}$  pin.

When latch-up occurs, power supply current increases rapidly and might thermally damage elements.

Also, take care to prevent the analog power supply voltage ( $AV_{cc}$ ) and analog input voltage from exceeding the digital power supply voltage ( $V_{cc}$ ) when the analog system power supply is turned on or off.

#### ● Stable Supply Voltage

Supply voltage should be stabilized.

A sudden change in power-supply voltage may cause a malfunction even within the guaranteed operating range of the  $V_{cc}$  power-supply voltage.

For stabilization, in principle, keep the variation in  $V_{cc}$  ripple (p-p value) in a commercial frequency range (50 Hz/60 Hz) not to exceed 10% of the standard  $V_{cc}$  value and suppress the voltage variation so that the transient variation rate does not exceed 0.1 V/ms during a momentary change such as when the power supply is switched.

#### ● Precautions for Use of External Clock

Even when an external clock is used, oscillation stabilization wait time is required for power-on reset, wake-up from sub clock mode or stop mode.

#### ● Serial Communication

There is a possibility to receive wrong data due to noise or other causes on the serial communication. Therefore, design a printed circuit board so as to avoid noise. Consider receiving of wrong data, for example, apply a checksum of data at the end to detect an error. If an error is detected, retransmit the data.

**■ Precautions for Debug**

When using an evaluation device (mounted on an MCU board) for software development, there may be some differences between the operation of the evaluation device and the device you will actually use. The following lists some points to note during development.

**● SYCC Register Settings**

During debugging, the values of the DIV1 and DIV0 bits in the SYCC register may differ from the user settings. This is because, when a break occurs, the CPU adjusts the communications speed between the evaluation device and the BGM adapter to use the optimum speed.

To prevent this from occurring, you need to set response speed optimization to disabled.

For this information, refer also to "2.3.1 Setting Operating Environment" in "F<sup>2</sup>MC-8L/8FX Family SOFTUNE Workbench USER'S MANUAL".

**● Flash Memory Types and Sizes**

Each evaluation device can be used for debugging of a number of different production models (series). When developing your program, please take note of the actual ROM and RAM sizes on the device you intend to use.

Further, evaluation devices use dual-operation flash memory. However, some production models have flash memory containing only one sector. Please take note of any differences between the flash memory configurations of the production and evaluation devices, particularly if writing a program that performs self-updating of flash memory.

**● Differences in Flash Memory Content**

The debugger for the F<sup>2</sup>MC-8FX family uses the software break instruction to implement break points. When continuous or step execution is performed after setting a break point, the software break instruction is written to the break address in the flash memory on the evaluation device.

Accordingly, the contents of flash memory after a software break has been inserted by the debugger will be different to the program data image generated by the compiler. Before performing a checksum, you must remember to clear all break points and "synchronize flash memory".

**● Restrictions Relating to the Flash Memory on the Evaluation Device**

The following restrictions apply to the evaluation device for the F<sup>2</sup>MC-8FX family.

- (1) Writing or erasing the lower bank (addresses 1000<sub>H</sub> to 3FFF<sub>H</sub>) is not possible.

When debugging, please do this on the production flash memory model.

- (2) Do not use the chip erase command for the flash memory on the evaluation device. When debugging, please do this on the production flash memory model.

● Operation of Peripheral Functions During a Break

When a CPU break occurs, the debugger for the F<sup>2</sup>MC-8FX family halts CPU operation (instruction code fetch, decoding, instruction execution, updating the PC, etc.) but the peripheral functions (PPG timer, UART, A/D converter, etc.) continue to operate.

The following are some example implications:

- (1) If the overflow flag for a timer/counter is set during a CPU break and the interrupt is enabled, the interrupt routine will run immediately when execution restarts after the break.
- (2) Clearing the overflow flag for a timer/counter via the memory window or similar during a CPU break will not appear to work as the flag will quickly be reset again.

● Prohibited Access to Undefined I/O Addresses

The debugger for the F<sup>2</sup>MC-8FX family uses the same evaluation device for debugging all models. This evaluation device includes all peripheral functions that may be used during debugging. Accessing a register that does not exist on your target production device may invoke a peripheral function that should not exist and may result in abnormal operation. Accordingly, please do not access undefined address areas.

#### ■ Pin Connection

##### ● Treatment of Unused Pin

Leaving unused input pins unconnected can cause abnormal operation or latch-up, leaving to permanent damage. Unused input pins should always be pulled up or down through resistance of at least 2 k $\Omega$ .

Any unused input/output pins may be set to output mode and left open, or set to input mode and treated the same as unused input pins. If there is unused output pin, make it open.

##### ● Treatment of Power Supply Pins on A/D Converter

Connect to be  $AV_{CC} = V_{CC}$  and  $AV_{SS} = V_{SS}$  even if the A/D converter is not in use.

Noise riding on the  $AV_{CC}$  pin may cause accuracy degradation. Therefore, it is recommended to connect approx. 0.1  $\mu$ F ceramic capacitor as a bypass capacitor between  $AV_{CC}$  and  $AV_{SS}$  pins in the vicinity of this device.

##### ● Power Supply Pins

In products with multiple  $V_{CC}$  or  $V_{SS}$  pins, the pins of the same potential are internally connected in the device to avoid abnormal operations including latch-up. However, you must connect the pins to external power supply and a ground line to lower the electro-magnetic emission level, to prevent abnormal operation of strobe signals caused by the rise in the ground level, and to conform to the total output current rating. Moreover, connect the current supply source with the  $V_{CC}$  and  $V_{SS}$  pins of this device at the low impedance.

It is also advisable to connect a ceramic bypass capacitor of approximately 0.1  $\mu$ F between  $V_{CC}$  and  $V_{SS}$  pins near this device.

##### ● Mode Pin (MOD)

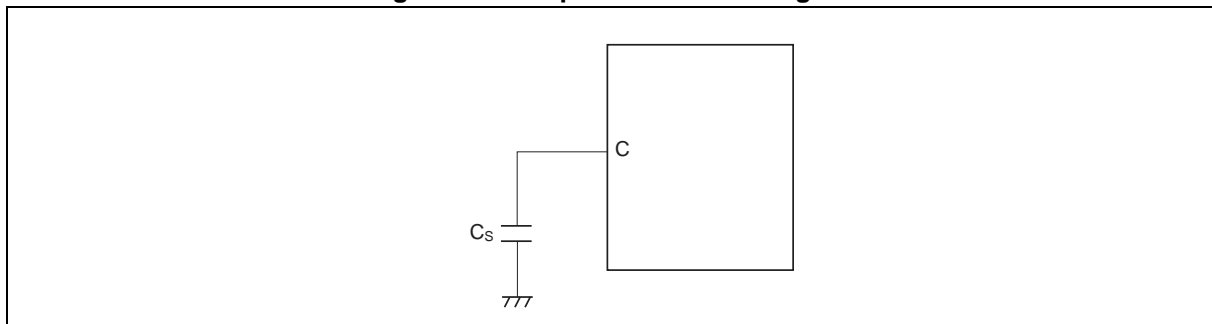
Connect the mode pin directly to  $V_{CC}$  or  $V_{SS}$ .

To prevent the device unintentionally entering test mode due to noise, lay out the printed circuit board so as to minimize the distance from the mode pins to  $V_{CC}$  or  $V_{SS}$  and to provide a low-impedance connection.

##### ● C pin

Use a ceramic capacitor or a capacitor with equivalent frequency characteristics. A bypass capacitor of  $V_{CC}$  pin must have a capacitance value higher than  $C_S$ . For connection of smoothing capacitor  $C_S$ , see Figure 2.1-1.

**Figure 2.1-1 C pin connection diagram**





- NC Pins

Any pins marked "NC" must be left open.

- Analog Power Supply

Always set the same potential to  $AV_{CC}$  and  $V_{CC}$ . When  $V_{CC} > AV_{CC}$ , the current may flow through analog input pins (AN).

# ***CHAPTER 3***

---

# ***MEMORY SPACE***

**This chapter describes memory space.**

3.1 Memory Space

3.2 Memory Map

## 3.1 Memory Space

---

The memory space on the F<sup>2</sup>MC-8FX family is 64 K bytes, divided into I/O, extended I/O, data, and program areas. The memory space includes special-purpose areas such as the general-purpose registers and vector table.

---

### ■ Configuration of Memory Space

#### ● I/O area (addresses: 0000<sub>H</sub> to 007F<sub>H</sub>)

- This area contains the control registers and data registers for on-chip peripheral resources.
- As the I/O area is allocated as part of memory space, it can be accessed in the same way as for memory. It can also be accessed at higher speed by using direct addressing instructions.

#### ● Extended I/O area (addresses: 0F80<sub>H</sub> to 0FFF<sub>H</sub>)

- This area contains the control registers and data registers for on-chip peripheral resources.
- As the extended I/O area is allocated as part of memory space, it can be accessed in the same way as for memory.

#### ● Data area

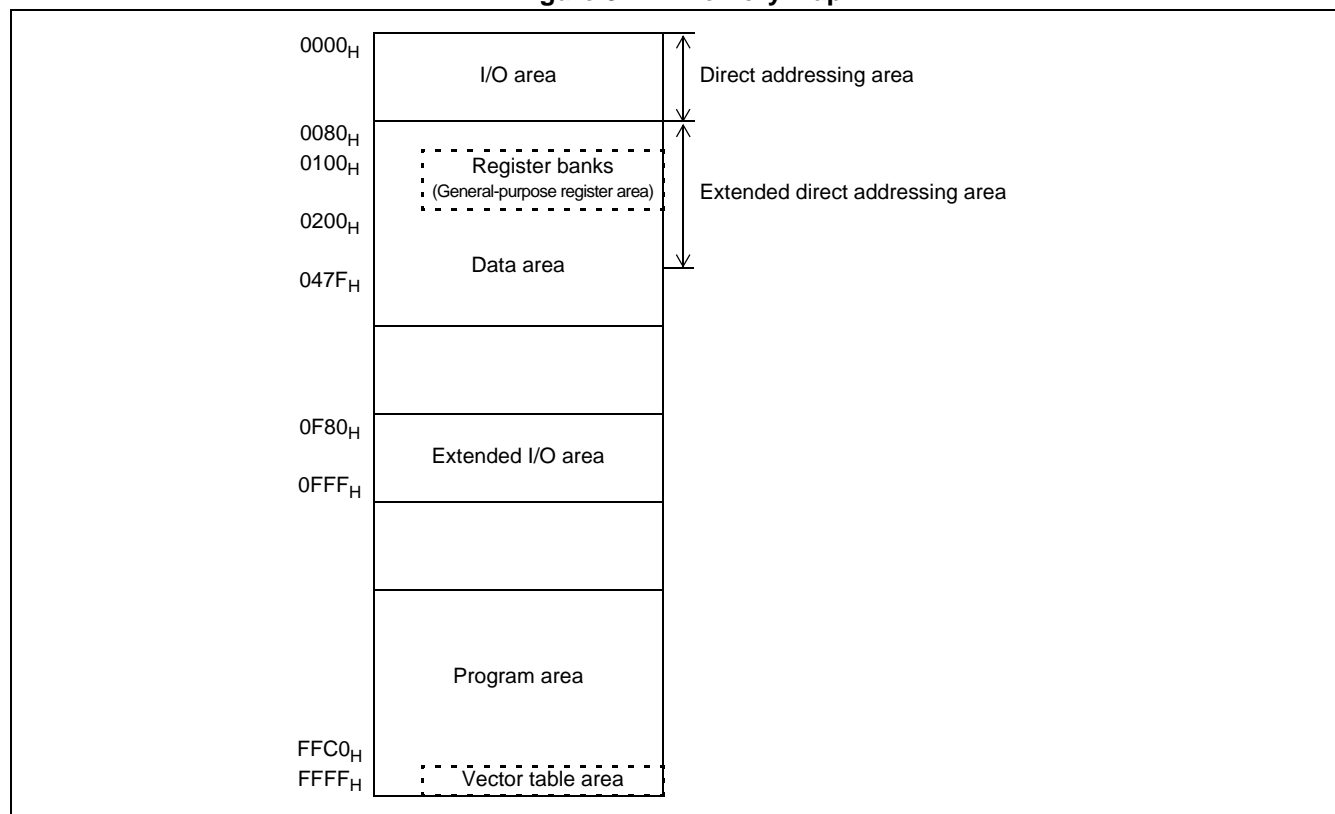
- Static RAM is incorporated as the internal data area.
- The internal RAM capacity is different depending on the product.
- The RAM area from 0000<sub>H</sub> to 007F<sub>H</sub> can be accessed at higher speed by using direct addressing instructions.
- The area from 0080<sub>H</sub> to 047F<sub>H</sub> is an extended direct addressing area. It can be accessed at higher speed by direct addressing instructions with the direct bank pointer set.
- Addresses 0100<sub>H</sub> to 01FF<sub>H</sub> can be used as a general-purpose register area.

#### ● Program area

- ROM is incorporated as the internal program area.
- The internal ROM capacity is different depending on the model.
- Addresses FFC0<sub>H</sub> to FFFF<sub>H</sub> are used as the vector table.

## ■ Memory Map

**Figure 3.1-1 Memory Map**



### 3.1.1 Areas for Specific Applications

---

The general-purpose register area and vector table area are used for the specific applications.

---

#### ■ General-purpose Register Area (Addresses: 0100<sub>H</sub> to 01FF<sub>H</sub>)

- This area contains the auxiliary registers used for 8-bit arithmetic or transfer operations.
- As the area is allocated as part of the RAM area, it can also be used as ordinary RAM.
- When the area is used as general-purpose registers, general-purpose register addressing enables higher-speed access using short instructions.

For details, see Section "5.1.1 Register Bank Pointer (RP)" and Section "5.2 General-purpose Registers".

#### ■ Vector Table Area (Addresses: FFC0<sub>H</sub> to FFFF<sub>H</sub>)

- This area is used as the vector table for vector call instructions (CALLV), interrupts, and resets.
- The vector table area is allocated at the top of the ROM area. At the individual addresses in the vector table, the start addresses of their respective service routines are set as data.

Table 8.1-1 lists the vector table addresses to be referenced for vector call instructions, interrupts, and for resets.

For details, see "CHAPTER 8 INTERRUPTS", "CHAPTER 7 RESET", and "■Special Instructions● CALLV #vct" in Appendix "E.2 Special Instruction".

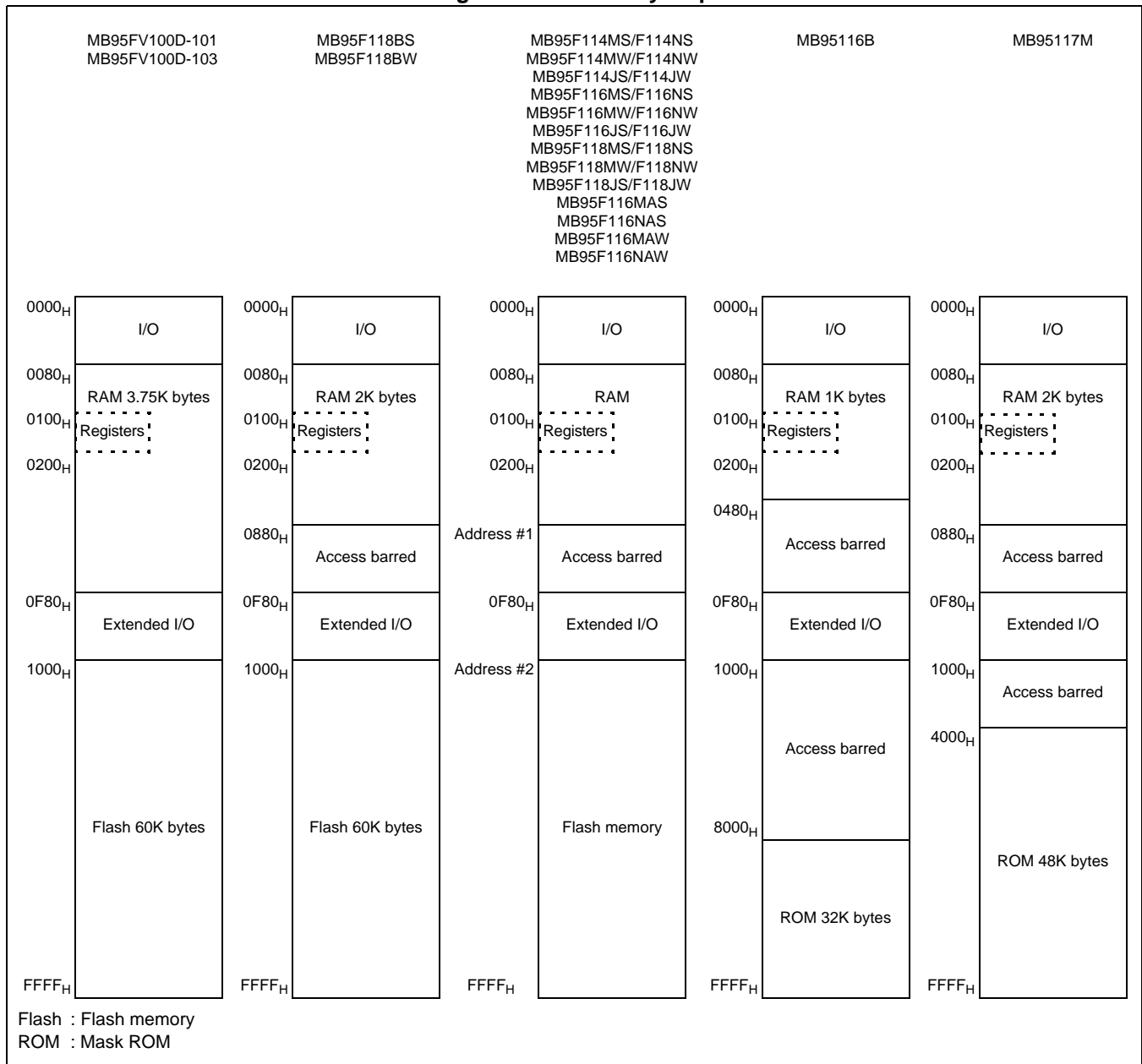
# MB95110B/M Series

## 3.2 Memory Map

This section gives a memory map of the MB95110B/M series.

### ■ Memory Map

Figure 3.2-1 Memory Map



	Flash memory	RAM	Address #1	Address #2
MB95F114MS/F114NS	16K bytes	512 bytes	0280 <sub>H</sub>	C000 <sub>H</sub>
MB95F114MW/F114NW				
MB95F114JS/F114JW				
MB95F116MS/F116NS	32K bytes	1K byte	0480 <sub>H</sub>	8000 <sub>H</sub>
MB95F116MW/F116NW				
MB95F116MAS/MB95F116NAS				
MB95F116MAW/MB95F116NAW				
MB95F116JS/F116JW				
MB95F118MS/F118NS	60K bytes	2K bytes	0880 <sub>H</sub>	1000 <sub>H</sub>
MB95F118MW/F118NW				
MB95F118JS/F118JW				

# ***CHAPTER 4***

---

# ***MEMORY ACCESS MODE***

**This chapter describes the memory access mode.**

## **4.1 Memory Access Mode**



## 4.1 Memory Access Mode

The memory access mode supported by this series is only single-chip mode.

### ■ Single-chip Mode

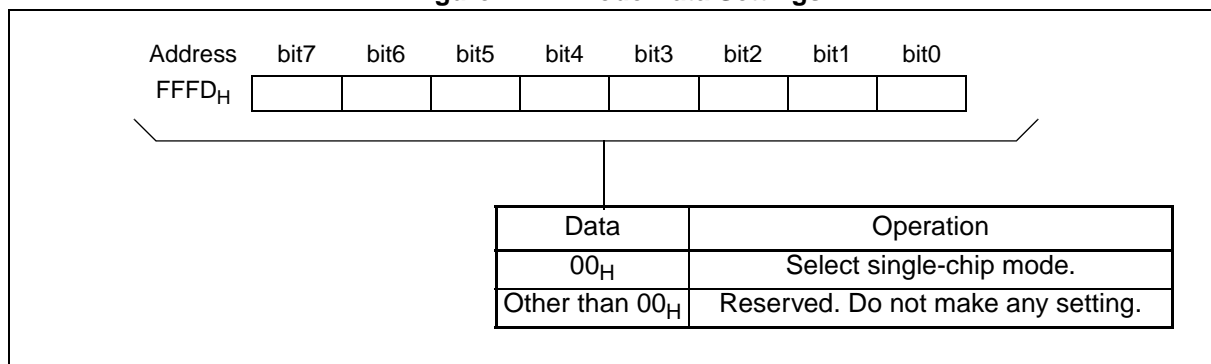
Single-chip mode uses only internal RAM and ROM. External bus access is not used.

#### ● Mode data

Mode data is used to determine the memory access mode of the CPU.

The mode data address is fixed as  $\text{FFFD}_{\text{H}}$  (the value of  $\text{FFFC}_{\text{H}}$  can be any value). Be sure to set the mode data of internal ROM to "00<sub>H</sub>" to select single-chip mode.

Figure 4.1-1 Mode Data Settings



After a reset, the CPU fetches mode data first.

The CPU then fetches the reset vector after the mode data. The instruction is performed from the address set by reset vector.

#### ● Mode pin (MOD)

Be sure to set the mode pin (MOD) to  $V_{\text{SS}}$ .

# ***CHAPTER 5***

---

## ***CPU***

**This chapter describes functions and operations of the CPU.**

5.1 Dedicated Registers

5.2 General-purpose Registers

5.3 Placement of 16-bit Data in Memory

## 5.1 Dedicated Registers

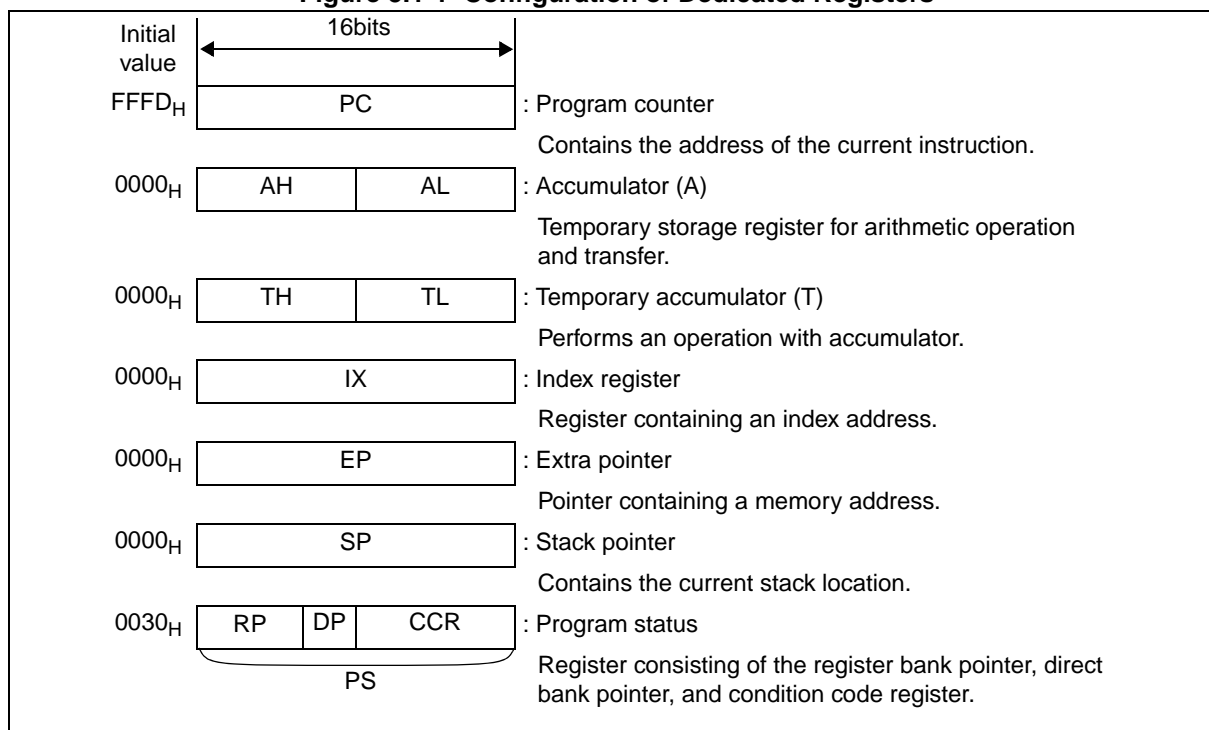
The CPU has its dedicated registers: the program counter (PC), two arithmetic registers (A and T), three address pointers (IX, EP, and SP), and the program status (PS) register. Each of the registers is 16 bits long. The PS register consists of the register bank pointer (RP), direct pointer (DP), and condition code register (CCR).

### ■ Configuration of Dedicated Registers

The dedicated registers in the CPU are seven 16-bit registers. Accumulator (A) and temporary accumulator (T) can also be used with only their lower eight bits in service.

Figure 5.1-1 shows the configuration of the dedicated registers.

**Figure 5.1-1 Configuration of Dedicated Registers**



### ■ Functions of Dedicated Registers

#### ● Program counter (PC)

The program counter is a 16-bit counter which contains the memory address of the instruction currently executed by the CPU. The program counter is updated whenever an instruction is executed or an interrupt or reset occurs. The initial value set immediately after a reset is the mode data read address (FFFD<sub>H</sub>).

#### ● Accumulator (A)

The accumulator is a 16-bit register for arithmetic operation. It is used for a variety of arithmetic and transfer operations of data in memory or data in other registers such as the temporary accumulator (T). The data in the accumulator can be handled either as word (16-bit) data or byte (8-bit) data. For byte-length arithmetic and transfer operations, only the lower eight bits (AL) of the accumulator are used with the upper eight bits (AH) left unchanged. The initial value after a reset is "0000<sub>H</sub>".

## ● Temporary accumulator (T)

The temporary accumulator is an auxiliary 16-bit register for arithmetic operation. It is used to perform arithmetic operations with the data in the accumulator (A). The data in the temporary accumulator is handled as word data for word-length (16-bit) operations with the accumulator (A) and as byte data for byte-length (8-bit) operations. For byte-length operations, only the lower eight bits (TL) of the temporary accumulator are used and the upper eight bits (TH) are not used.

When a MOV instruction is used to transfer data to the accumulator (A), the previous contents of the accumulator are automatically transferred to the temporary accumulator. When transferring byte-length data, the upper eight bits (TH) of the temporary accumulator remain unchanged. The initial value after a reset is "0000<sub>H</sub>".

## ● Index register (IX)

The index register is a 16-bit register used to hold the index address. The index register is used with a single-byte offset (-128 to +127). The offset value is added to the index address to generate the memory address for data access. The initial value after a reset is "0000<sub>H</sub>".

## ● Extra pointer (EP)

The extra pointer is a 16-bit register which contains the value indicating the memory address for data access. The initial value after a reset is "0000<sub>H</sub>".

## ● Stack pointer (SP)

The stack pointer is a 16-bit register which holds the address referenced when an interrupt or subroutine call occurs and by the stack push and pop instructions. During program execution, the value of the stack pointer indicates the address of the most recent data pushed onto the stack. The initial value after a reset is "0000<sub>H</sub>".

## ● Program status (PS)

The program status is a 16-bit control register. The upper eight bits make up the register bank pointer (RP) and direct bank pointer (DP); the lower eight bits make up the condition code register (CCR).

In the upper eight bits, the upper five bits make up the register bank pointer used to contain the address of the general-purpose register bank. The lower three bits make up the direct bank pointer which locates the area to be accessed at high speed by direct addressing.

The lower eight bits make up the condition code register (CCR) which consists of flags that represent the state of the CPU.

The instructions that can access the program status are MOVW A,PS or MOVW PS,A. The register bank pointer (RP) and direct bank pointer (DP) in the program status register can also be read from or written to by accessing the mirror address (0078<sub>H</sub>).

Note that the condition code register (CCR) is part of the program status register and cannot be accessed independently.

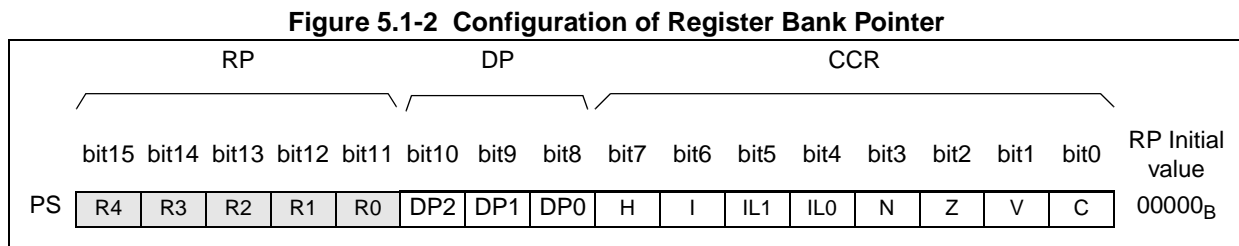
Refer to the "F<sup>2</sup>MC-8FX Programming Manual" for details on using the dedicated registers.

## 5.1.1 Register Bank Pointer (RP)

The register bank pointer (RP) in bits 15 to 11 of the program status (PS) register contains the address of the general-purpose register bank that is currently in use and is translated into a real address when general-purpose register addressing is used.

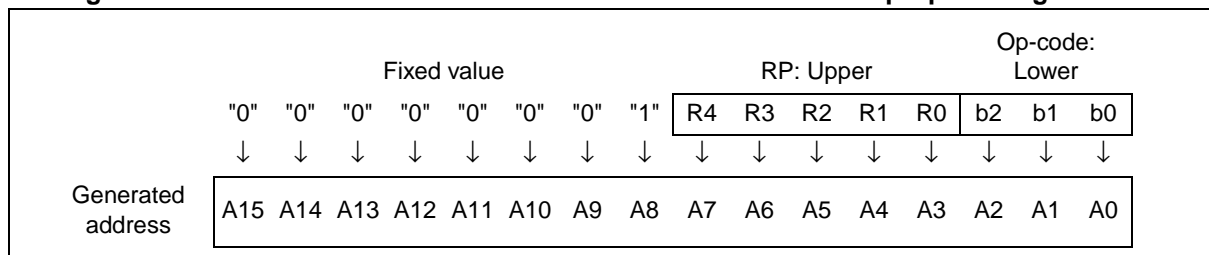
### ■ Configuration of Register Bank Pointer (RP)

Figure 5.1-2 shows the configuration of the register bank pointer.



The register bank pointer contains the address of the register bank currently being used. The content of the register bank pointer is translated into a real address according to the rule shown in Figure 5.1-3.

**Figure 5.1-3 Rule for Translation into Real Addresses in General-purpose Register Area**



The register bank pointer specifies the register bank used as general-purpose registers in the RAM area. There are a total of 32 register banks. The current register bank is specified by setting a value between 0 and 31 in the upper five bits of the register bank pointer. Each register bank has eight 8-bit general-purpose registers which are selected by the lower three bits of the op-code.

The register bank pointer allows the space from "0100<sub>H</sub>" to up to "01FF<sub>H</sub>" to be used as a general-purpose register area. Note, however, that the available area is limited depending on the product. The initial value after a reset is "0000<sub>H</sub>".

### ■ Mirror Address for Register Bank and Direct Bank Pointers

The register bank pointer (RP) and direct bank pointer (DP) can be written to and read from by accessing the program status (PS) register using the "MOVW A,PS" and "MOVW PS,A" instructions, respectively. They can also be written to and read from directly by accessing mirror address "0078<sub>H</sub>" of the register bank pointer.

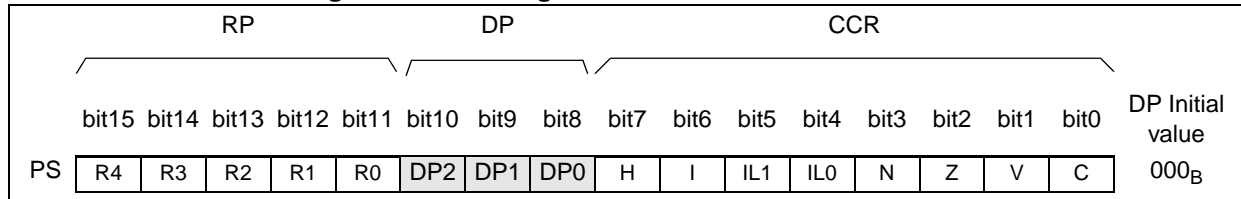
## 5.1.2 Direct Bank Pointer (DP)

The direct bank pointer (DP) in bits 10 to 8 of the program status (PS) register specifies the area to be accessed by direct addressing.

### ■ Configuration of Direct Bank Pointer (DP)

Figure 5.1-4 shows the configuration of the direct bank pointer.

**Figure 5.1-4 Configuration of Direct Bank Pointer**



The areas from 0000<sub>H</sub> to 007F<sub>H</sub> and 0080<sub>H</sub> to 047F<sub>H</sub> can be accessed by direct addressing. Access to 0000<sub>H</sub> to 007F<sub>H</sub> is specified with an operand regardless of the value in the direct bank pointer. Access to 0080<sub>H</sub> to 047F<sub>H</sub> is specified with the value in the value of the direct bank pointer and the operand.

Table 5.1-1 shows the relationship between direct bank pointer (DP) and access area; Table 5.1-2 lists the direct addressing instructions.

**Table 5.1-1 Direct Access Pointer and Access Area**

Direct bank pointer (DP) [2:0]	Operand-specified dir	Access area
XXX <sub>B</sub> (It does not affect the mapping.)	0000 <sub>H</sub> to 007F <sub>H</sub>	0000 <sub>H</sub> to 007F <sub>H</sub>
000 <sub>B</sub> (Initial value)	0080 <sub>H</sub> to 00FF <sub>H</sub>	0080 <sub>H</sub> to 00FF <sub>H</sub>
001 <sub>B</sub>		0100 <sub>H</sub> to 017F <sub>H</sub>
010 <sub>B</sub>		0180 <sub>H</sub> to 01FF <sub>H</sub>
011 <sub>B</sub>		0200 <sub>H</sub> to 027F <sub>H</sub>
100 <sub>B</sub>		0280 <sub>H</sub> to 02FF <sub>H</sub>
101 <sub>B</sub>		0300 <sub>H</sub> to 037F <sub>H</sub>
110 <sub>B</sub>		0380 <sub>H</sub> to 03FF <sub>H</sub>
111 <sub>B</sub>		0400 <sub>H</sub> to 047F <sub>H</sub>

**Table 5.1-2 Direct Address Instruction List**

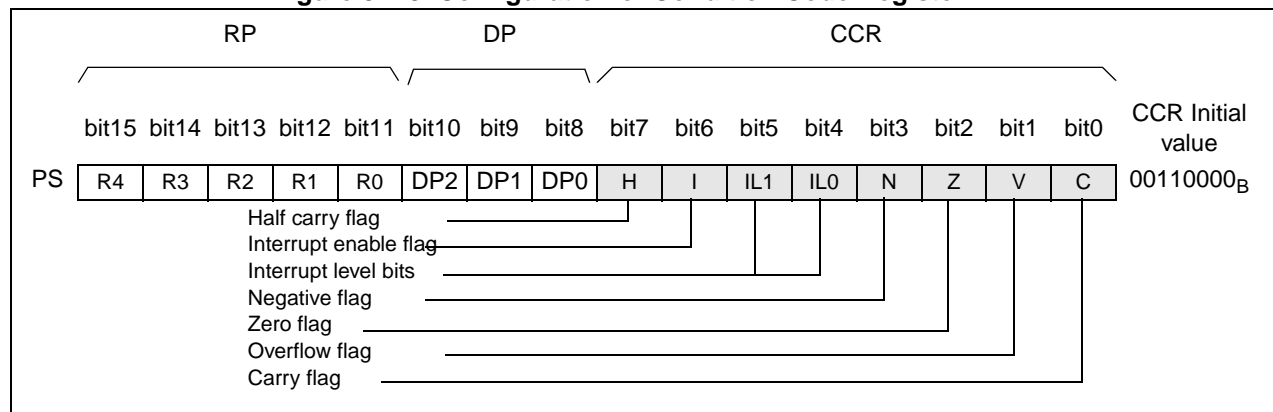
Applicable Instruction
CLRB dir:bit
SETB dir:bit
BBC dir:bit,rel
BBS dir:bit,rel
MOV A,dir
CMP A,dir
ADDC A,dir
SUBC A,dir
MOV dir,A
XOR A,dir
AND A,dir
OR A,dir
MOV dir,#imm
CMP dir,#imm
MOVW A,dir
MOVW dir,A

### 5.1.3 Condition Code Register (CCR)

The condition code register (CCR) in the lower eight bits of the program status (PS) register consists of the bits (H, N, Z, V, and C) containing information about the arithmetic result or transfer data and the bits (I, IL1, and IL0) used to control the acceptance of interrupt requests.

#### ■ Configuration of Condition Code Register (CCR)

Figure 5.1-5 Configuration of Condition Code Register



The condition code register is a part of the program status (PS) register and therefore cannot be accessed independently.

#### ■ Bits Result Information Bits

##### ● Half carry flag (H)

This flag is set to "1" when a carry from bit3 to bit4 or a borrow from bit4 to bit3 occurs as the result of an operation. Otherwise, the flag is set to "0". Do not use this flag for any operation other than addition and subtraction as the flag is intended for decimal-adjusted instructions.

##### ● Negative flag (N)

This flag is set to "1" when the value of the most significant bit is "1" as the result of an operation and set to "0" if the value is "0".

##### ● Zero flag (Z)

This flag is set to "1" when the result of an operation is "0" and set to "0" otherwise.

##### ● Overflow flag (V)

This flag indicates whether an operation has resulted in an overflow, assuming the operand used for the operation as an integer represented by a two's complement. The flag is set to "1" when an overflow occurs and set to "0" otherwise.

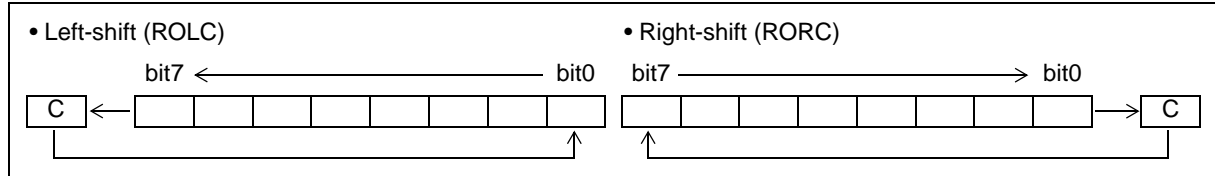


● Carry flag (C)

This flag is set to "1" when a carry from bit7 or a borrow to bit7 occurs as the result of an operation. Otherwise, the flag is set to "0". When a shift instruction is executed, the flag is set to the shift-out value.

Figure 5.1-6 shows how the carry flag is updated by a shift instruction.

**Figure 5.1-6 Carry Flag Updated by Shift Instruction**



■ Interrupt Acceptance Control Bits

● Interrupt enable flag (I)

When this flag is set to "1", interrupts are enabled and accepted by the CPU. When this flag is set to "0", interrupts are disabled and rejected by the CPU.

The initial value after a reset is "0".

The SETI and CLRI instructions set and clear the flag to "1" and "0", respectively.

● Interrupt level bits (IL1, IL0)

These bits indicate the level of the interrupt currently accepted by the CPU.

The interrupt level is compared with the value of the interrupt level setting register (ILR0 to ILR5) that corresponds to the interrupt request (IRQ0 to IRQ23) of each peripheral resource.

The CPU services an interrupt request only when its interrupt level is smaller than the value of these bits with the interrupt enable flag set (CCR: I = 1). Table 5.1-3 lists interrupt level priorities. The initial value after a reset is "11<sub>B</sub>".

**Table 5.1-3 Interrupt Levels**

IL1	IL0	Interrupt Level	Priority
0	0	0	High
0	1	1	<div style="text-align: center;">           ↑            ↓         </div>
1	0	2	
1	1	3	
			Low (No interrupt)

The interrupt level bits (IL1, IL0) are usually "11<sub>B</sub>" with the CPU not servicing an interrupt (with the main program running).

For details on interrupts, see Section "8.1 Interrupts".

## 5.2 General-purpose Registers

The general-purpose registers are memory blocks consisting of eight 8-bit registers per bank. A total of up to 32 register banks can be used. The register bank pointer (RP) is used to specify the register bank.

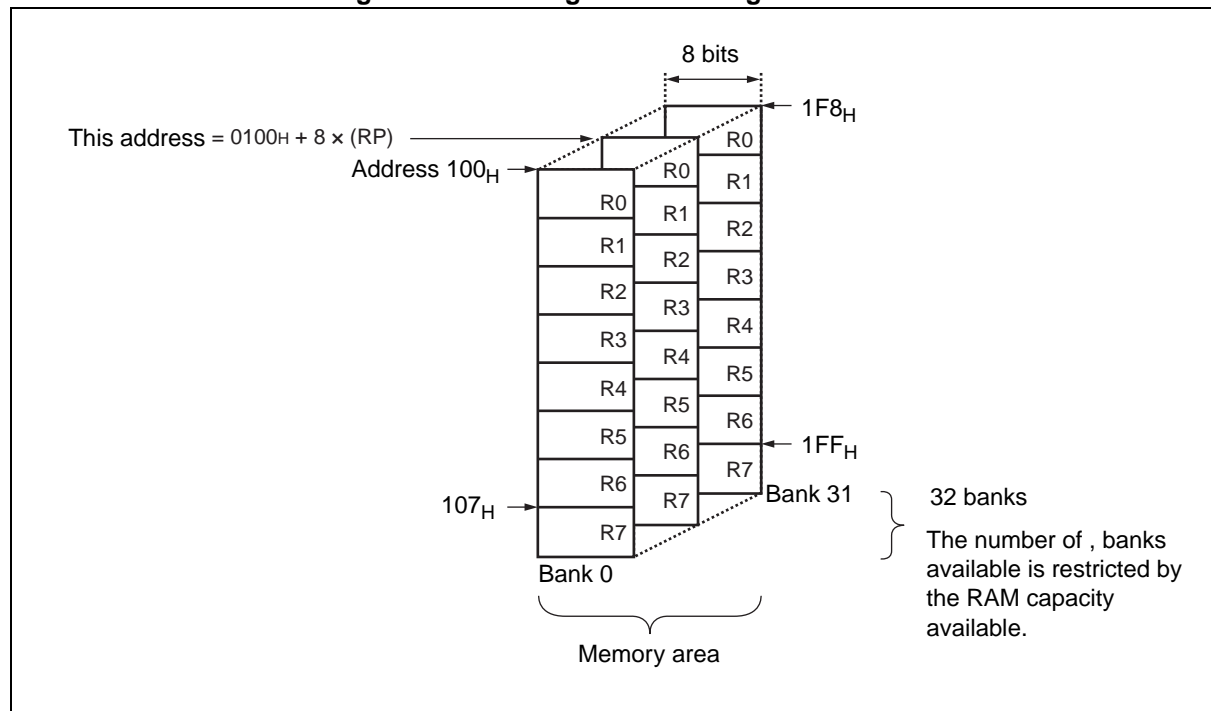
Register banks are useful for interrupt handling, vector call processing, and subroutine calls.

### ■ Configuration of General-purpose Registers

- The general-purpose registers are 8-bit registers and are located in register banks in the general-purpose register area (in RAM).
- Up to 32 banks can be used, where each bank consists of eight registers (R0 to R7).
- The register bank pointer (RP) specifies the register bank currently being used and the lower three bits of the op-code specify general-purpose register 0 (R0) to 7 (R7).

Figure 5.2-1 shows the configuration of the register banks.

**Figure 5.2-1 Configuration of Register Banks**



For information on the general-purpose register area available on each model, see Section "3.1.1 Areas for Specific Applications".

## ■ Features of General-purpose Registers

There are the following features in the general-purpose registers:

- High-speed access to RAM using short instructions (general-purpose register addressing).
- Blocks of register banks facilitating data backup and division by function unit.

General-purpose register banks can be allocated exclusively for specific interrupt service routines or vector call (CALLV #0 to #7) processing routines. An example is always using the fourth register bank for the second interrupt.

Only specifying a dedicated register bank at the beginning of an interrupt service routine automatically saves the general-purpose registers before the interrupt. This eliminates the need for pushing general-purpose register data onto the stack, allowing the CPU to accept interrupts at high speed.

---

### Notes:

When coding an interrupt service routine, be careful not to change the value of the interrupt level bits (CCR: IL1, IL0) in the condition code register when specifying the register bank by updating the register bank pointer (RP) in that routine. Perform the programming by using either of them.

- Read the interrupt level bits and save their value before writing to the RP.
  - Directly write to the RP mirror address "0078<sub>H</sub>" to update the RP.
-

## 5.3 Placement of 16-bit Data in Memory

This section describes how 16-bit data is stored in memory.

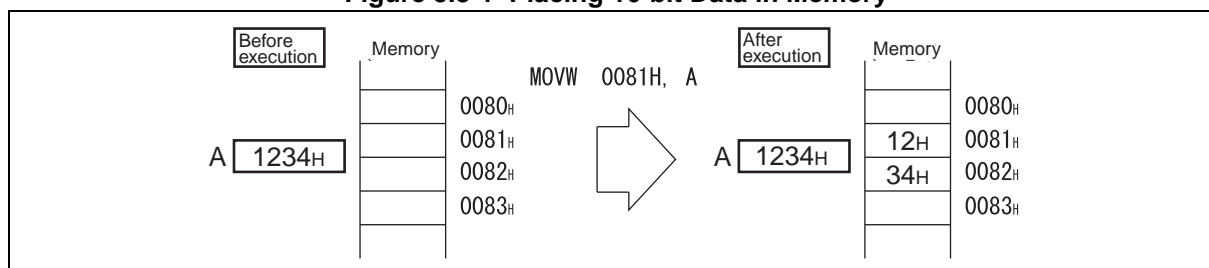
### ■ Placement of 16-bit Data in Memory

#### ● State of 16-bit data stored in RAM

When you write 16-bit data to memory, the upper byte of the data is stored at a smaller address and the lower byte is stored at the next address. When you read 16-bit data, it is handled in the same way.

Figure 5.3-1 shows how 16-bit data is placed in memory.

**Figure 5.3-1 Placing 16-bit Data in Memory**



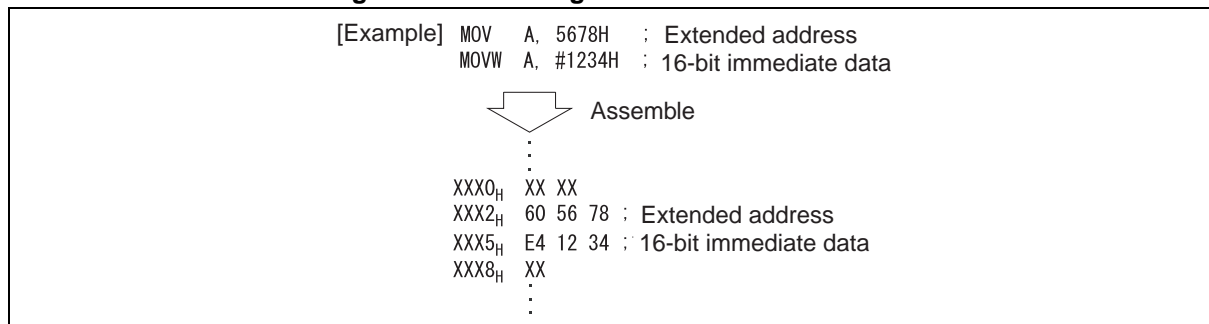
#### ● State of operand-specified 16-bit data

In the same way, even when the operands in an instruction specifies 16-bit data, the upper byte is stored at the address closer to the op-code (instruction) and the lower byte is stored at the next address.

That is true whether the operands are either memory addresses or 16-bit immediate data.

Figure 5.3-2 shows how 16-bit data in an instruction is placed.

**Figure 5.3-2 Storing 16-bit Data in Instruction**



#### ● State of 16-bit data in the stack

When 16-bit register data is pushed onto the stack upon an interrupt, the upper byte is stored at a lower address in the same way.



# CHAPTER 6

---

# CLOCK CONTROLLER

**This chapter describes the functions and operations of the clock controller.**

- 6.1 Overview of Clock Controller
- 6.2 Oscillation Stabilization Wait Time
- 6.3 System Clock Control Register (SYCC)
- 6.4 PLL Control Register (PLLC)
- 6.5 Oscillation Stabilization Wait Time Setting Register (WATR)
- 6.6 Standby Control Register (STBC)
- 6.7 Clock Modes
- 6.8 Operations in Low-power Consumption Modes (Standby Modes)
- 6.9 Clock Oscillator Circuits
- 6.10 Overview of Prescaler
- 6.11 Configuration of Prescaler
- 6.12 Operating Explanation of Prescaler
- 6.13 Notes on Use of Prescaler

## **6.1 Overview of Clock Controller**

---

**The F<sup>2</sup>MC-8FX family has a built-in clock controller that optimizes its power consumption. It includes dual clock product supporting both of the main clock and sub clock and single clock product supporting only the main clock.**

**The clock controller enables/disables clock oscillation, enables/disables the supply of clock signals to the internal circuitry, selects the clock source, and controls the PLL and frequency divider circuits.**

---

### **■ Overview of Clock Controller**

The clock controller enables/disables clock oscillation, enables/disables clock supply to the internal circuitry, selects the clock source, and controls the PLL and frequency divider circuits.

The clock controller controls the internal clock according to the clock mode, standby mode settings and the reset operation. The current clock mode selects the internal operating clock and the standby mode selects whether to enable or disable clock oscillation and signal supply.

The clock controller selects the optimum power consumption and features depending on the combination of clock mode and standby mode.

Dual clock product have four different source clocks: a main clock, which is the main oscillation clock divided by two, a sub clock, which is the sub oscillation clock divided by two, a main PLL clock, which is the main oscillation clock multiplied by the PLL multiplier. However, for MB95F116MAW/NAW, they do not have sub PLL clock.

Single clock product have two different source clocks: a main clock, which is the main oscillation clock divided by two; and a main PLL clock, which is the main oscillation clock multiplied by the PLL multiplier.





The clock controller consists of the following blocks:

- Main clock oscillator circuit

This block is the oscillator circuit for the main clock.

- Sub clock oscillator circuit (Dual clock product)

This block is the oscillator circuit for the sub clock.

- Main PLL oscillator circuit

This block is the oscillator circuit for the main PLL.

- Sub PLL oscillator circuit (Dual clock product, except MB95F116MAW/F116NAW)

This block is the oscillator circuit for the sub PLL clock.

- System clock selector

This block selects one of the four different source clocks for main clock, sub clock, main PLL clock, and sub PLL clock depending on the clock mode. The prescaler frequency-divides the selected source clock into the machine clock. It is supplied to the clock control circuit.

- Clock control circuit

This block controls the supply of the machine clock to the CPU and each peripheral resource according to the standby mode or oscillation stabilization wait time.

- Oscillation stabilization wait circuit

This block outputs the oscillation stabilization wait time signal for each clock from 14 types of main clock oscillation stabilization signals created by the time-base timer and 15 types of sub clock oscillation stabilization signals created by the watch prescaler.

- System clock control register (SYCC)

This register is used to control current clock mode display, clock mode selection, machine clock divide ratio selection, and sub clock oscillation in main clock mode and main PLL clock mode.

- Standby control register (STBC)

This register is used to control the transition from RUN state to standby mode, the setting of pin states in stop mode, time-base timer mode, or watch mode, and the generation of software resets.

- PLL control register (PLLC)

This register is used to enable/disable the oscillation of the main PLL and sub PLL clocks, set the multiplier, and to indicate the stability of PLL oscillation.

- Oscillation stabilization wait time setting register (WATR)

This register is used to set the oscillation stabilization wait time for the main clock and sub clock.

#### ■ Clock Modes

There are four clock modes available: main clock mode, main PLL clock mode, sub clock mode, and sub PLL clock mode.

Table 6.1-1 shows the relationships between the clock modes and the machine clock (operating clock for the CPU and peripheral resources).

**Table 6.1-1 Clock Modes and Machine Clock Selection**

Clock Mode	Machine Clock
Main clock mode	The machine clock is generated from the main clock (main clock divided by 2).
Main PLL clock mode	The machine clock is generated from the main PLL clock (main clock multiplied by the PLL multiplier).
Sub clock mode (Dual clock product only)	The machine clock is generated from the sub clock (sub clock divided by 2).
Sub PLL clock mode (Dual clock product only, except MB95F116MAW/F116NAW)	The machine clock is generated from the sub PLL clock (sub clock multiplied by the PLL multiplier).

In any of the clock modes, the selected clock can also be frequency-divided. Additionally, in modes using a PLL clock, a multiplier for the clock frequency can also be set.

#### ■ Peripheral Resources not Affected by Clock Mode

Note that the peripheral resources listed in the table below are not affected by the clock mode, division, and PLL multiplier settings. Table 6.1-2 lists the peripheral resources not affected by the clock mode.

**Table 6.1-2 Peripheral Resources Not Affected by Clock Mode**

Peripheral Resource	Operating Clock
Time-base timer	Main clock ( $2^1/F_{CH}$ : main clock divided by 2)
Watchdog timer	Main clock (with time-base timer output selected) Sub clock (with watch prescaler output selected) (dual clock product only)
Watch prescaler (Dual clock product only)	Sub clock ( $2^1/F_{CL}$ : sub clock divided by 2)
Clock counter (Dual clock product only)	Sub clock (watch prescaler output)

For some peripheral resources other than those listed above, it may be possible to select the time-base timer or watch prescaler output as a count clock. Check the description of each peripheral resource for details.

## ■ Standby Modes

The clock controller selects whether to enable or disable clock oscillation and clock supply to internal circuitry depending on each standby mode. With the exception of time-base timer mode and watch mode, the standby mode can be set independently of the clock mode.

Table 6.1-3 shows the relationships between standby modes and clock supply states.

**Table 6.1-3 Standby Modes and Clock Supply States**

Standby Mode	Clock Supply States
Sleep mode	Stops clock supply to the CPU and watchdog timer. As a result, the CPU stops operation, but other peripheral resources continue operating.
Time-base timer mode	Supplies clock signals only to the time-base timer, watch prescaler, and watch counter while stopping clock supply to other circuits. As a result, all the functions other than the time-base timer, watch prescaler, watch counter, external interrupt, and low-voltage detection reset (option) are stopped. time-base timer mode is only the standby mode for main clock mode or main PLL clock mode.
Watch mode (Dual clock product only)	Stops main clock oscillation, but supplies clock signals only to the watch prescaler and watch counter while stopping clock supply to other circuits. As a result, all the functions other than the watch prescaler, watch counter, external interrupt, and low-voltage detection reset (option) are stopped. Watch mode is only the standby mode for sub clock mode or sub PLL clock mode.
Stop mode	Stops main clock oscillation and sub clock oscillation and stops the supply of all clock signals. As a result, all the functions other than external interrupt and low-voltage detection reset (option) are stopped.

#### ■ Combinations of Clock Mode and Standby Mode

Table 6.1-4 lists the combinations of clock mode and standby mode and their respective operating states of internal circuits.

**Table 6.1-4 Combinations of Standby Mode and Clock Mode and Internal Operating States**

Function	RUN				Sleep				Time-base timer		Watch (Dual clock product)		Stop	
	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product, except MB95F116MAW/F116NAW)	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product, except MB95F116MAW/F116NAW)	Main clock mode	Main PLL clock mode	Sub clock mode (Dual clock product)	Sub PLL clock mode (Dual clock product, except MB95F116MAW/F116NAW)	Main (PLL) clock mode	Sub PLL clock mode (Dual clock product, except MB95F116MAW/F116NAW)
Main clock	Operating		Stopped		Operating		Stopped		Operating		Stopped		Stopped	Stopped
Main PLL clock	Stopped*1	Operating	Stopped		Stopped*1	Operating	Stopped		Stopped*1		Stopped		Stopped	Stopped
Sub clock	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operating*2	Stopped
Sub PLL clock	Stopped*3		Stopped*3	Operating	Stopped*3		Stopped*3	Operating	Stopped*3		Stopped*3	Operating	Stopped*3	Stopped
CPU	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	Stopped
ROM	Operating		Operating		Value held		Value held		Value held		Value held		Value held	Value held
RAM														
I/O ports	Operating		Operating		Output held		Output held		Output held		Output held		Output held/Hi-Z	Output held/Hi-Z
Time-base timer	Operating		Stopped		Operating		Stopped		Operating		Stopped		Stopped	Stopped
Watch prescaler	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operating*2	Stopped
Watch counter	Operating*2		Operating		Operating*2		Operating		Operating*2		Operating		Operating*4	Stopped
External interrupt	Operating		Operating		Operating		Operating		Operating		Operating		Operating	Operating
Watchdog timer	Operating		Operating		Stopped		Stopped		Stopped		Stopped		Stopped	Stopped
Low-voltage detection reset	Operating		Operating		Operating		Operating		Operating		Operating		Operating	Operating
Other peripheral resources	Operating		Operating		Operating		Operating		Stopped		Stopped		Stopped	Stopped

\*1: Operates when the main PLL clock oscillation enable bit in the PLL control register (PLLC:MPEN) is set to "1".

\*2: Stops when the sub clock oscillation stop bit in the system clock control register (SYCC:SUSBS) is set to "1".

\*3: Operates when the sub PLL clock oscillation enable bit in the PLL control register (PLLC:SPEN) is set to "1".

\*4: Watch counter keeps counting and no interrupts occur. When the sub clock oscillation stop bit in the system clock control register (SYCC: SUSBS) is set to "1", watch counter stops.

## 6.2 Oscillation Stabilization Wait Time

The oscillation stabilization wait time is the time after the oscillator circuit stops oscillation until the oscillator resumes its stable oscillation at its natural frequency. The clock controller obtains the oscillation stabilization wait time by counting a set number of oscillation clock cycles to prevent clock supply to internal circuits.

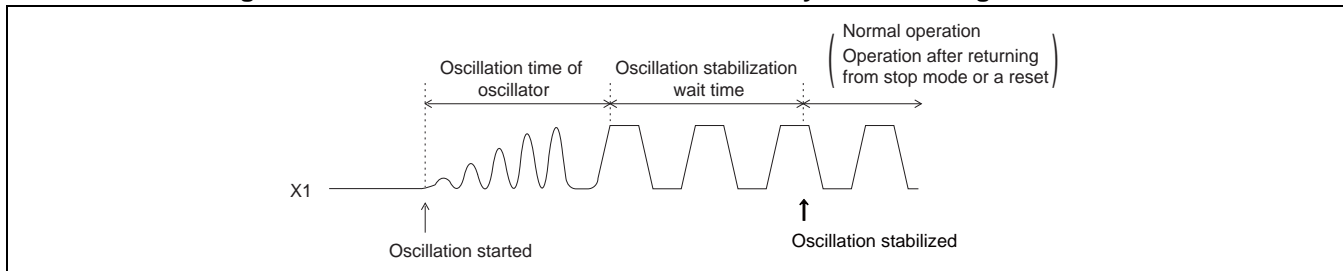
### ■ Oscillation Stabilization Wait Time

The clock controller obtains the oscillation stabilization wait time followed by the initiation of oscillation by counting a set number of oscillation clock cycles to prevent clock supply to internal circuits.

When a state transition request for starting oscillation when the power is turned on or for restarting halted oscillation at a clock mode change by a reset, an interrupt in standby mode, or by software, the clock controller automatically waits until the oscillation stabilization wait time for the main clock or sub clock has passed and then causes transition to the next state.

Figure 6.2-1 shows oscillation immediately after being started.

**Figure 6.2-1 Behavior of Oscillator Immediately after Starting Oscillation**



The main clock oscillation stabilization wait time is counted by using the time-base timer. The sub clock oscillation stabilization wait time is counted by using the watch prescaler. The count can be set in the oscillation stabilization wait time setting register (WATR). Set it in keeping with the oscillator characteristics.

When a power-on reset occurs, the oscillation stabilization wait time is fixed to the initial value. For masked ROM products, however, you can specify the initial value of the oscillation stabilization wait time when ordering masked ROM.

Table 6.2-1 shows the length of oscillation stabilization wait time.

**Table 6.2-1 Oscillation Stabilization Wait Time**

Clock	Factor	Oscillation Stabilization Wait Time
Main clock	Power-on reset	Initial value: $(2^{14}-2)/F_{CH}$ , where $F_{CH}$ is the main clock frequency (specified when ROM is ordered for mask ROM products)
	Other than power-on reset	Register setting value (WATR:MWT3, MWT2, MWT1, MWT0)
Sub clock (Dual clock product)	Power-on reset	Initial value: $(2^{15}-2)/F_{CL}$ , where $F_{CL}$ is the sub clock frequency.
	Other than power-on reset	Register setting value (WATR:SWT3, SWT2, SWT1, SWT0)

After the oscillation stabilization wait time of the main clock ends, the oscillation stabilization wait time of sub clock measurement is begun.

## ■ PLL Clock Oscillation Stabilization Wait Time

As with the oscillation stabilization wait time of the oscillator, the clock controller automatically waits for the PLL oscillation stabilization wait time to elapse after a request for state transition from PLL oscillation stopped state to oscillation start is generated via an interrupt in standby mode or a change of clock mode by software.

Note that the PLL clock oscillation stabilization wait time changes according to the PLL startup timing.

Table 6.2-2 shows the PLL oscillation stabilization wait time.

**Table 6.2-2 PLL Oscillation Stabilization Wait Time**

	PLL Oscillation Stabilization Wait Time		Remarks
	Minimum time	Maximum time	
Main PLL clock	$2^{11}/F_{CH} \times 2$	$2^{11}/F_{CH} \times 3$	<ul style="list-style-type: none"> <li>Oscillation stabilization wait time is taken while <math>2^{11}/F_{CH}</math> is counted twice (minimum) or three times (maximum).</li> <li><math>F_{CH}</math> represents the main clock frequency.</li> </ul>
Sub PLL clock (Dual clock product, except MB95F116MAW/ F116NAW)	$2^8/F_{CL} \times 2$	$2^8/F_{CL} \times 3$	<ul style="list-style-type: none"> <li>Oscillation stabilization wait time is taken while <math>2^8/F_{CL}</math> is counted twice (minimum) or three times (maximum).</li> <li><math>F_{CL}</math> represents the sub clock frequency.</li> </ul>

## ■ Oscillation Stabilization Wait Time and Clock Mode/Standby Mode Transition

The clock controller automatically waits for the oscillation stabilization wait time to elapse as needed when the operating state causes a transition. Depending on the state transition, however, the clock controller does not always wait for the oscillation stabilization wait time.

For details on state transitions, see "6.7 Clock Modes" and "6.8 Operations in Low-power Consumption Modes (Standby Modes)".

## 6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) is used to indicate and switch the current clock mode, select the machine clock divide ratio, and control sub clock oscillation in main clock mode and main PLL clock mode.

### ■ Configuration of System Clock Control Register (SYCC)

Figure 6.3-1 Configuration of System Clock Control Register (SYCC)

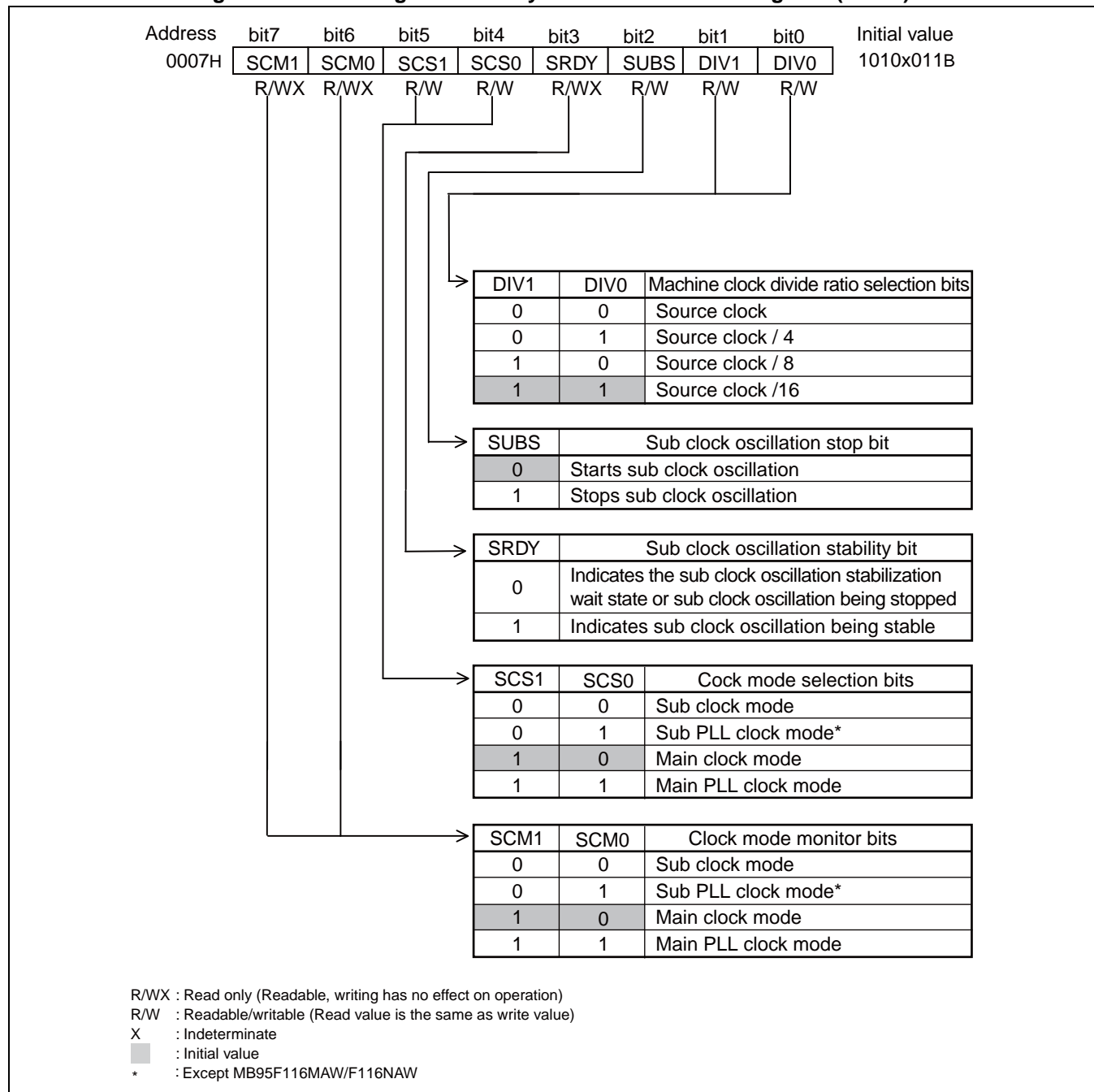


Table 6.3-1 Functions of Bits in System Clock Control Register (SYCC)

Bit name		Function																				
bit7, bit6	SCM1, SCM0: Clock mode monitor bits	Indicate the current clock mode. <b>When set to "00<sub>B</sub>"</b> : the bits indicate sub clock mode. <b>When set to "01<sub>B</sub>"</b> : the bit indicate sub PLL clock mode (except MB95F116MAW/F116NAW). <b>When set to "10<sub>B</sub>"</b> : the bit indicate main clock mode. <b>When set to "11<sub>B</sub>"</b> : the bit indicate main PLL clock mode. These bits are read-only; any writing is meaningless.																				
bit5, bit4	SCS1, SCS0: Clock mode selection bits	Specify the clock mode. <b>When set to "00<sub>B</sub>"</b> : the bits specify transition to sub clock mode (dual clock product only). <b>When set to "01<sub>B</sub>"</b> : the bits specify transition to sub PLL clock mode (dual clock product only except MB95F116MAW/F116NAW). <b>When set to "10<sub>B</sub>"</b> : the bits specify transition to main clock mode. <b>When set to "11<sub>B</sub>"</b> : the bits specify transition to main PLL clock mode. Once a clock mode has been selected in the SCS1 and SCS0 bits, any attempt to write to them is ignored until the transition to that clock mode is completed. On single clock product, an attempt to write "00 <sub>B</sub> " or "01 <sub>B</sub> " to these bits is ignored, leaving their value unchanged.																				
bit3	SRDY: Sub clock oscillation stability bit (Dual clock product only)	Indicates whether sub clock oscillation has become stable. <ul style="list-style-type: none"><li>When set to "1", the SRDY bit indicates that the oscillation stabilization wait time for the sub clock has passed.</li><li>When set to "0", the SRDY bit indicates that the clock controller is in the sub clock oscillation stabilization wait state or that sub clock oscillation has been stopped.</li></ul> This bit is read-only; any writing is meaningless. On single clock product, the value of the bit is meaningless.																				
bit2	SUBS: Sub clock oscillation stop bit (Dual clock product only)	Stops sub clock oscillation in main clock mode or main PLL clock mode. <b>When set to "0"</b> : the bit enables sub clock oscillation. <b>When set to "1"</b> : the bit stops sub clock oscillation. Notes: <ul style="list-style-type: none"><li>In sub clock mode or sub PLL clock mode, the sub clock oscillates regardless of the value of this bit, except in stop mode.</li><li>In main clock mode or main PLL clock mode as well, the sub clock oscillates regardless of the value of this bit when sub PLL clock oscillation has been enabled by the PLL clock oscillation enable bit in the PLL control register (PLLC:SPEN).</li><li>Do not update the SYCC: SCS1 bit and this bit at the same time.</li><li>On single clock product, the value of this bit has no effect on operation.</li></ul>																				
bit1, bit0	DIV1, DIV0: Machine clock divide ratio selection bits	<ul style="list-style-type: none"><li>These bits select the machine clock divide ratio to the source clock.</li><li>The machine clock is generated from the source clock according to the divide ratio set by the bits.</li></ul> <table><tr><th>DIV1</th><th>DIV0</th><th>Machine Clock Divide Ratio Selection Bits</th><th>SCM1, SCM0 = 10<sub>B</sub></th></tr><tr><td>0</td><td>0</td><td>Source clock (No division)</td><td>Main clock divided by 2</td></tr><tr><td>0</td><td>1</td><td>Source clock/4</td><td>Main clock divided by 8</td></tr><tr><td>1</td><td>0</td><td>Source clock/8</td><td>Main clock divided by 16</td></tr><tr><td>1</td><td>1</td><td>Source clock/16</td><td>Main clock divided by 32</td></tr></table>	DIV1	DIV0	Machine Clock Divide Ratio Selection Bits	SCM1, SCM0 = 10 <sub>B</sub>	0	0	Source clock (No division)	Main clock divided by 2	0	1	Source clock/4	Main clock divided by 8	1	0	Source clock/8	Main clock divided by 16	1	1	Source clock/16	Main clock divided by 32
DIV1	DIV0	Machine Clock Divide Ratio Selection Bits	SCM1, SCM0 = 10 <sub>B</sub>																			
0	0	Source clock (No division)	Main clock divided by 2																			
0	1	Source clock/4	Main clock divided by 8																			
1	0	Source clock/8	Main clock divided by 16																			
1	1	Source clock/16	Main clock divided by 32																			



## 6.4 PLL Control Register (PLLC)

The PLL control register (PLLC) controls the main PLL clock and sub PLL clock.

### ■ Configuration of PLL Control Register (PLLC)

Figure 6.4-1 Configuration of PLL Control Register (PLLC)

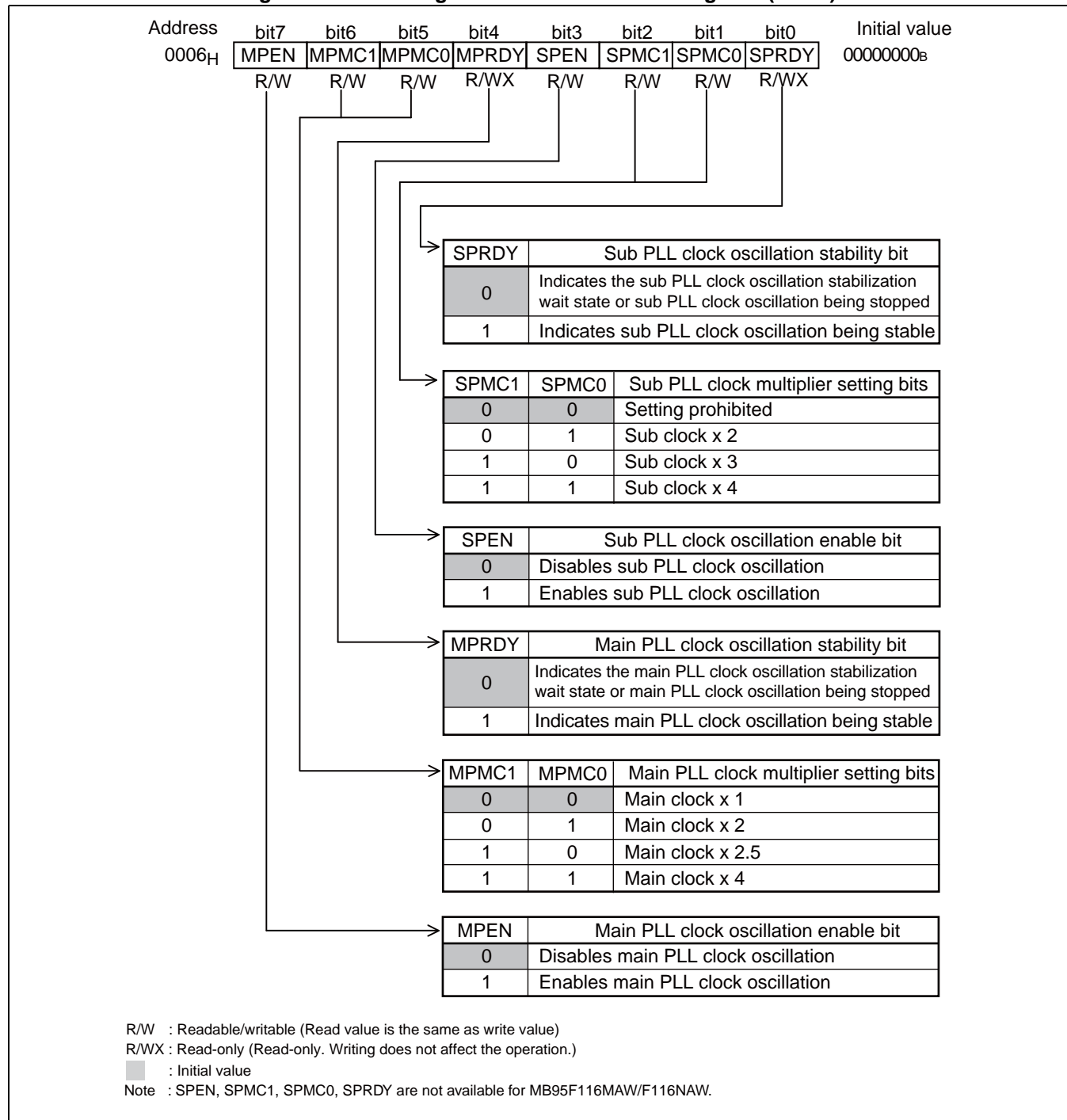


Table 6.4-1 Functions of Bits in PLL Control Register (PLLC) (1 / 2)

Bit name		Function															
bit7	MPEN: Main PLL clock oscillation enable bit	<p>Enables or disables the oscillation of the main PLL clock in main clock mode or time-base timer mode.</p> <p><b>When set to "0":</b> the bit disables main PLL clock oscillation.</p> <p><b>When set to "1":</b> the bit enables main PLL clock oscillation.</p> <p>In main PLL clock mode, the main PLL clock oscillates regardless of the value of this bit either in the RUN state or in sleep mode.</p>															
bit6, bit5	MPMC1, MPMC0: Main PLL clock multiplier setting bits	<p>Set the multiplier for the main PLL clock.</p> <table border="1"> <thead> <tr> <th>MPMC1</th><th>MPMC0</th><th>Main PLL clock multiplier setting bits</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Main clock <math>\times</math> 1</td></tr> <tr> <td>0</td><td>1</td><td>Main clock <math>\times</math> 2</td></tr> <tr> <td>1</td><td>0</td><td>Main clock <math>\times</math> 2.5</td></tr> <tr> <td>1</td><td>1</td><td>Main clock <math>\times</math> 4</td></tr> </tbody> </table> <p>Note: The value of these bits can be changed only when the main PLL clock is stopped. Therefore, do not attempt to update the bits with the PLL clock oscillation enable bit (MPEN) is set to "1" or with the clock mode selection bits in the system clock control register (SYCC: SCS1, SCS0) are set to "11<sub>B</sub>". (It is however possible to set these bits at the same time as setting MPEN to "1".)</p>	MPMC1	MPMC0	Main PLL clock multiplier setting bits	0	0	Main clock $\times$ 1	0	1	Main clock $\times$ 2	1	0	Main clock $\times$ 2.5	1	1	Main clock $\times$ 4
MPMC1	MPMC0	Main PLL clock multiplier setting bits															
0	0	Main clock $\times$ 1															
0	1	Main clock $\times$ 2															
1	0	Main clock $\times$ 2.5															
1	1	Main clock $\times$ 4															
bit4	MPRDY: Main PLL clock oscillation stability bit	<p>Indicates whether main PLL clock oscillation has become stable.</p> <ul style="list-style-type: none"> <li>When set to "1", the MPRDY bit indicates that the oscillation stabilization wait time for the main PLL clock has passed.</li> <li>When set to "0", the MPRDY bit indicates that the clock controller is in the main PLL clock oscillation stabilization wait state or that main PLL clock oscillation has been stopped.</li> </ul> <p>This bit is read-only; any writing is meaningless and has no effect on the operation.</p>															
bit3	SPEN: Sub PLL clock oscillation enable bit (Dual clock product only except MB95F116MAW/F116NAW)	<p>Enables or disables the oscillation of the sub PLL clock in main clock mode, main PLL clock mode, sub clock mode, or in watch mode.</p> <p><b>When set to "0":</b> the bit disables sub PLL clock oscillation.</p> <p><b>When set to "1":</b> the bit enables sub PLL clock oscillation.</p> <p>In sub PLL clock mode, the sub PLL clock oscillates regardless of the value of this bit except in watch mode.</p> <p>Even in sub PLL clock mode, the sub PLL clock stops oscillation in stop mode regardless of the value of this bit.</p> <p>On single clock product, the value of the bit has no effect on the operation.</p>															

Table 6.4-1 Functions of Bits in PLL Control Register (PLLC) (2 / 2)

Bit name		Function															
bit2, bit1	SPMC1, SPMC0: Sub PLL clock multiplier setting bits (Dual clock product only except MB95F116MAW/F116NAW )	<p>Set the multiplier for the Sub PLL clock.</p> <table border="1"> <thead> <tr> <th>SPMC1</th><th>SPMC0</th><th>Sub PLL Clock Multiplier Setting Bits</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Setting prohibited. Be sure to write any other value before using the PLL.</td></tr> <tr> <td>0</td><td>1</td><td>sub clock × 2</td></tr> <tr> <td>1</td><td>0</td><td>sub clock × 3</td></tr> <tr> <td>1</td><td>1</td><td>sub clock × 4</td></tr> </tbody> </table> <p>On single clock product, the value of these bits has no effect on the operation.</p> <p>Notes:</p> <ul style="list-style-type: none"> <li>Although the initial value of these bits is "00<sub>B</sub>", the PLL does not operate normally with this setting. Be sure to set the bits to any value other than "00<sub>B</sub>" either before setting the sub PLL clock oscillation enable bit (SPEN) to "1" or before setting the clock mode selection bits in the system clock control register (SYCC:SCS1, SCS0) to "01<sub>B</sub>".</li> <li>These bits can be updated only when the sub PLL clock is stopped. Consequently, you should not update the bits either with the sub PLL clock oscillation enable bit (SPEN) set to "1" or with the system clock selection bits in the system clock control register (SYCC:SCS1, SCS0) set to "01<sub>B</sub>". (It is however possible to set these bits at the same time as setting SPEN to "1".)</li> </ul>	SPMC1	SPMC0	Sub PLL Clock Multiplier Setting Bits	0	0	Setting prohibited. Be sure to write any other value before using the PLL.	0	1	sub clock × 2	1	0	sub clock × 3	1	1	sub clock × 4
SPMC1	SPMC0	Sub PLL Clock Multiplier Setting Bits															
0	0	Setting prohibited. Be sure to write any other value before using the PLL.															
0	1	sub clock × 2															
1	0	sub clock × 3															
1	1	sub clock × 4															
bit0	SPRDY: Sub PLL clock oscillation stability bit (Dual clock product only except MB95F116MAW/F116NAW )	<p>Indicates whether sub PLL clock oscillation has become stable.</p> <ul style="list-style-type: none"> <li>When set to "1", the SPRDY bit indicates that the oscillation stabilization wait time for the sub PLL clock has passed.</li> <li>When set to "0", the SPRDY bit indicates that the clock controller is in the sub PLL clock oscillation stabilization wait state or that sub PLL clock oscillation has been stopped.</li> </ul> <p>This bit is read-only; any writing is meaningless.</p> <p>On single clock product, the value of the bit is meaningless.</p>															

## 6.5 Oscillation Stabilization Wait Time Setting Register (WATR)

This register is used to set the oscillation stabilization wait time.

### ■ Configuration of Oscillation Stabilization Wait Time Setting Register (WATR)

Figure 6.5-1 Configuration of Oscillation Stabilization Wait Time Setting Register (WATR)

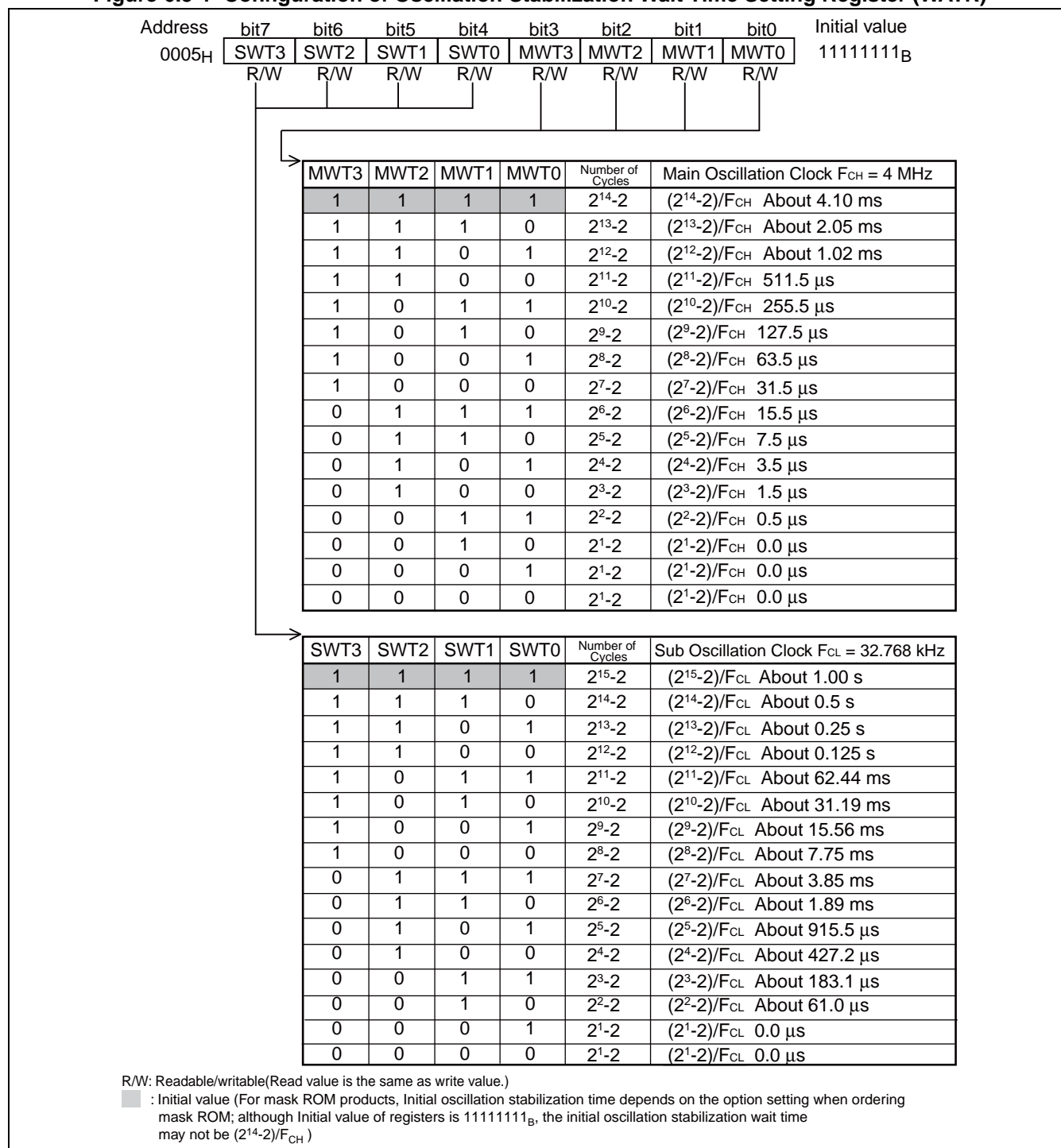


Table 6.5-1 Functions of Bits in Oscillation Stabilization Wait Time Setting Register (WATR) (1 / 2)

Bit name		Function		
bit7 to bit4	SWT3, SWT2, SWT1, SWT0: Sub clock oscillation stabilization wait time selection bits	Set the sub clock oscillation stabilization wait time.		
		SWT3 SWT2 SWT1 SWT0	Number of Cycles	Sub clock $F_{CL} = 32.768 \text{ kHz}$
		1111 <sub>B</sub>	$2^{15}-2$	$(2^{15}-2)/F_{CL}$ About 1.0 s
		1110 <sub>B</sub>	$2^{14}-2$	$(2^{14}-2)/F_{CL}$ About 0.5 s
		1101 <sub>B</sub>	$2^{13}-2$	$(2^{13}-2)/F_{CL}$ About 0.25 s
		1100 <sub>B</sub>	$2^{12}-2$	$(2^{12}-2)/F_{CL}$ About 0.125 s
		1011 <sub>B</sub>	$2^{11}-2$	$(2^{11}-2)/F_{CL}$ About 62.44 ms
		1010 <sub>B</sub>	$2^{10}-2$	$(2^{10}-2)/F_{CL}$ About 31.19 ms
		1001 <sub>B</sub>	$2^9-2$	$(2^9-2)/F_{CL}$ About 15.56 ms
		1000 <sub>B</sub>	$2^8-2$	$(2^8-2)/F_{CL}$ About 7.75 ms
		0111 <sub>B</sub>	$2^7-2$	$(2^7-2)/F_{CL}$ About 3.85 ms
		0110 <sub>B</sub>	$2^6-2$	$(2^6-2)/F_{CL}$ About 1.89 ms
		0101 <sub>B</sub>	$2^5-2$	$(2^5-2)/F_{CL}$ About 915.5 $\mu\text{s}$
		0100 <sub>B</sub>	$2^4-2$	$(2^4-2)/F_{CL}$ About 427.2 $\mu\text{s}$
		0011 <sub>B</sub>	$2^3-2$	$(2^3-2)/F_{CL}$ About 183.1 $\mu\text{s}$
		0010 <sub>B</sub>	$2^2-2$	$(2^2-2)/F_{CL}$ About 61.0 $\mu\text{s}$
		0001 <sub>B</sub>	$2^1-2$	$(2^1-2)/F_{CL}$ 0.0 $\mu\text{s}$
		0000 <sub>B</sub>	$2^1-2$	$(2^1-2)/F_{CL}$ 0.0 $\mu\text{s}$
		On single clock product, the value of these bits is meaningless.		
		Number of cycles in the above table is for a minimum value. Above-mentioned cycle number + $1/F_{CL}$ to the number of cycle in the above table for a maximum value.		
		Note: Do not update these bits during sub clock oscillation stabilization wait time. You should update them either with the sub clock oscillation stability bit in the system clock control register (SYCC:SRDY) set to "1" or in sub clock mode or sub PLL clock mode. You can also update them while the sub clock is stopped with the sub clock oscillation stop bit in the system clock control register (SYCC:SUS) set to "1" in main clock mode or main PLL clock mode.		

**Table 6.5-1 Functions of Bits in Oscillation Stabilization Wait Time Setting Register (WATR) (2 / 2)**

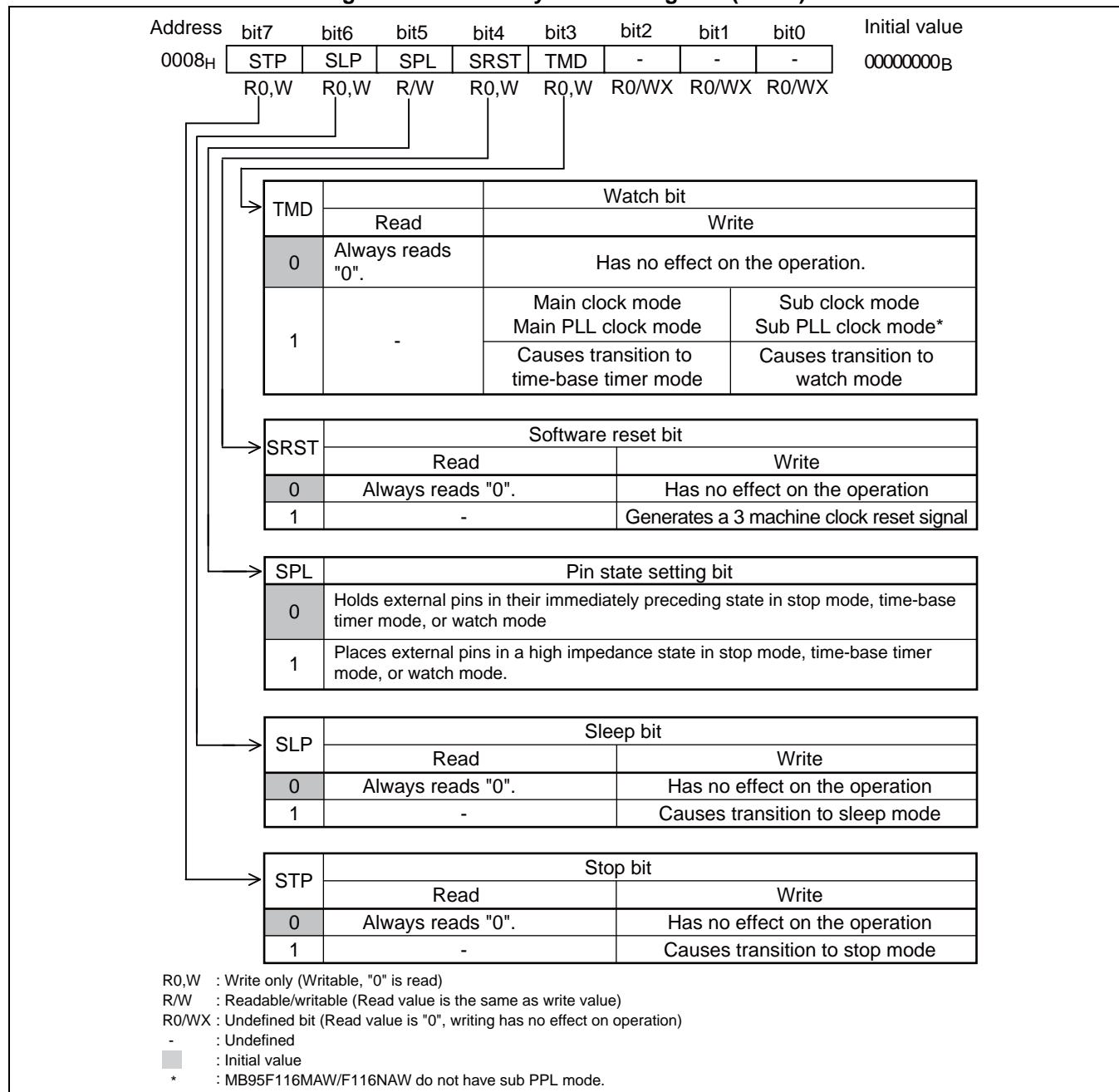
Bit name		Function		
bit3 to bit0	MWT3, MWT2, MWT1, MWT0: Main clock oscillation stabilization wait time selection bits	Set the main clock oscillation stabilization wait time.		
		MWT3 MWT2 MWT1 MWT0	Number of Cycles	Main clock F <sub>CH</sub> = 4 MHz
		1111 <sub>B</sub>	2 <sup>14</sup> -2	(2 <sup>14</sup> -2)/F <sub>CH</sub> About 4.10 ms
		1110 <sub>B</sub>	2 <sup>13</sup> -2	(2 <sup>13</sup> -2)/F <sub>CH</sub> About 2.05 ms
		1101 <sub>B</sub>	2 <sup>12</sup> -2	(2 <sup>12</sup> -2)/F <sub>CH</sub> About 1.02 ms
		1100 <sub>B</sub>	2 <sup>11</sup> -2	(2 <sup>11</sup> -2)/F <sub>CH</sub> 511.5 μs
		1011 <sub>B</sub>	2 <sup>10</sup> -2	(2 <sup>10</sup> -2)/F <sub>CH</sub> 255.5 μs
		1010 <sub>B</sub>	2 <sup>9</sup> -2	(2 <sup>9</sup> -2)/F <sub>CH</sub> 127.5 μs
		1001 <sub>B</sub>	2 <sup>8</sup> -2	(2 <sup>8</sup> -2)/F <sub>CH</sub> 63.5 μs
		1000 <sub>B</sub>	2 <sup>7</sup> -2	(2 <sup>7</sup> -2)/F <sub>CH</sub> 31.5 μs
		0111 <sub>B</sub>	2 <sup>6</sup> -2	(2 <sup>6</sup> -2)/F <sub>CH</sub> 15.5 μs
		0110 <sub>B</sub>	2 <sup>5</sup> -2	(2 <sup>5</sup> -2)/F <sub>CH</sub> 7.5 μs
		0101 <sub>B</sub>	2 <sup>4</sup> -2	(2 <sup>4</sup> -2)/F <sub>CH</sub> 3.5 μs
		0100 <sub>B</sub>	2 <sup>3</sup> -2	(2 <sup>3</sup> -2)/F <sub>CH</sub> 1.5 μs
		0011 <sub>B</sub>	2 <sup>2</sup> -2	(2 <sup>2</sup> -2)/F <sub>CH</sub> 0.5 μs
		0010 <sub>B</sub>	2 <sup>1</sup> -2	(2 <sup>1</sup> -2)/F <sub>CH</sub> 0.0 μs
		0001 <sub>B</sub>	2 <sup>1</sup> -2	(2 <sup>1</sup> -2)/F <sub>CH</sub> 0.0 μs
		0000 <sub>B</sub>	2 <sup>1</sup> -2	(2 <sup>1</sup> -2)/F <sub>CH</sub> 0.0 μs
Number of cycles is for a minimum value. Add 1/F <sub>CH</sub> to the minimum value for a maximum value.				
Note:    Do not update these bits during main clock oscillation stabilization wait time. You should update them in main clock mode or main PLL clock mode. You can also update them in sub clock mode.				

## 6.6 Standby Control Register (STBC)

The standby control register (STBC) is used to control transition from the RUN state to sleep mode, stop mode, time-base timer mode, or watch mode, set the pin state in stop mode, time-base timer mode, and watch mode, and to control the generation of software resets.

### ■ Standby Control Register (STBC)

Figure 6.6-1 Standby Control Register (STBC)



**Table 6.6-1 Functions of Bits in Standby Control Register (STBC)**

Bit name		Function
bit7	STP: Stop bit	Sets transition to stop mode. <b>When set to "0"</b> : the bit is meaningless. <b>When set to "1"</b> : the bit causes transition to stop mode. When read, the bit always returns "0". Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see "6.8.1 Notes on Using Standby Mode".
bit6	SLP: Sleep bit	Sets transition to sleep mode. <b>When set to "0"</b> : the bit is meaningless. <b>When set to "1"</b> : the bit causes transition to sleep mode. When read, the bit always returns "0". Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see "6.8.1 Notes on Using Standby Mode".
bit5	SPL: Pin state setting bit	Sets the states of external pins in stop mode, time-base timer mode, and watch mode. <b>When set to "0"</b> : the bit holds the states (levels) of external pins in stop mode, time-base timer mode, and watch mode. <b>When set to "1"</b> : the bit places external pins in a high impedance state in stop mode, time-base timer mode, and watch mode. (Those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register.)
bit4	SRST: Software reset bit	Sets a software reset. <b>When set to "0"</b> : the bit is meaningless. <b>When set to "1"</b> : the bit generates a 3 machine clock reset signal. When read, the bit always returns "0".
bit3	TMD: Watch bit	On dual clock product, this bit sets transition to time-base timer mode or watch mode. On single clock product, the bit sets transition to time-base timer mode. <ul style="list-style-type: none"> <li>Writing "1" to the bit in main clock mode or main PLL clock mode causes transition to time-base timer mode.</li> <li>Writing "1" to the bit in sub clock mode or sub PLL clock mode (except MB95F116MAW/F116NAW) causes transition to watch mode.</li> <li>Writing "0" to the bit is meaningless.</li> <li>When read, the bit always returns "0".</li> </ul> Note: An attempt to write "1" to this bit is ignored if an interrupt request has been issued. For details, see "6.8.1 Notes on Using Standby Mode".
bit2 to bit0	Undefined bits	When read, these bits always return "0". These are undefined bits. The bits are read-only; any writing is meaningless.



---

Notes:

- Set the standby mode after making sure that the transition to clock mode has been completed by comparing the values of the clock mode monitor bits (SYCC:SCM1,SCM0) and clock mode setting bits (SYCC:SCS1,SCS0) in the system clock control register.
- If you write "1" simultaneously to two or more of the stop bit (STP), sleep bit (SLP), software reset bit (SRST), and watch bit (TMD), priority is given to them in the following order:
  - (1) Software reset bit (SRST)
  - (2) Stop bit (STP)
  - (3) Watch bit (TMD)
  - (4) Sleep bit (SLP)

When released from the standby mode, the device returns to the normal operating status.

---

## MB95110B/M Series

### 6.7 Clock Modes

---

The clock modes available are: main clock mode, sub clock mode, main PLL clock mode, and sub PLL clock mode. Mode switching takes place according to the settings in the system clock control register (SYCC).

Sub clock mode and sub PLL clock mode are not supported by single clock product.

Sub PPL clock mode is not supported by dual clock products MB95F116MAW/F116NAW.

---

#### ■ Operations in Main Clock Mode

Main clock mode uses the main clock as the machine clock for the CPU and peripheral resources.

The time-base timer operates with the main clock.

The watch prescaler and watch counter operate with the sub clock (on dual clock product).

If you set standby mode during operation in main clock mode, the device can enter sleep mode, stop mode, or time-base timer mode.

After a reset, main clock mode is always set regardless of the clock mode used before the reset.

#### ■ Operations in Sub Clock Mode (on Dual Clock Product)

Sub clock mode uses the sub clock as the machine clock for the CPU and peripheral resources with main clock oscillation stopped. In this mode, the time-base timer remains stopped as it requires the main clock for operation.

If you set standby mode during operation in sub clock mode, the device can enter sleep mode, stop mode, or watch mode.

#### ■ Operations in Main PLL Clock Mode

Main PLL clock mode uses the main PLL clock as the machine clock for the CPU and peripheral resources.

The time-base timer and watchdog timer operate with the main clock.

The watch prescaler and watch counter operate with the sub clock (on dual clock product).

If you set standby mode during operation in main PLL clock mode, the device can enter sleep mode, stop mode, or time-base timer mode.

#### ■ Operations in Sub PLL Clock Mode (on Dual Clock Product, except MB95F116MAW/F116NAW)

Sub PLL clock mode uses the sub PLL clock as the machine clock for the CPU and peripheral resources with main clock oscillation stopped. In this mode, the time-base timer remains stopped as it requires the main clock for operation. The watch prescaler and watch counter operate with the sub clock.

If you set standby mode during operation in sub PLL clock mode, the device can enter sleep mode, stop mode, or watch mode.

### ■ Clock Mode State Transition Diagram

The clock modes available are: main clock mode, main PLL clock mode, sub clock mode, and sub PLL clock mode. The device can switch between these modes according to the settings in the system clock control register (SYCC).

**Figure 6.7-1 Clock Mode State Transition Diagram (Dual Clock Product, except MB95F116MAW/F116NAW)**

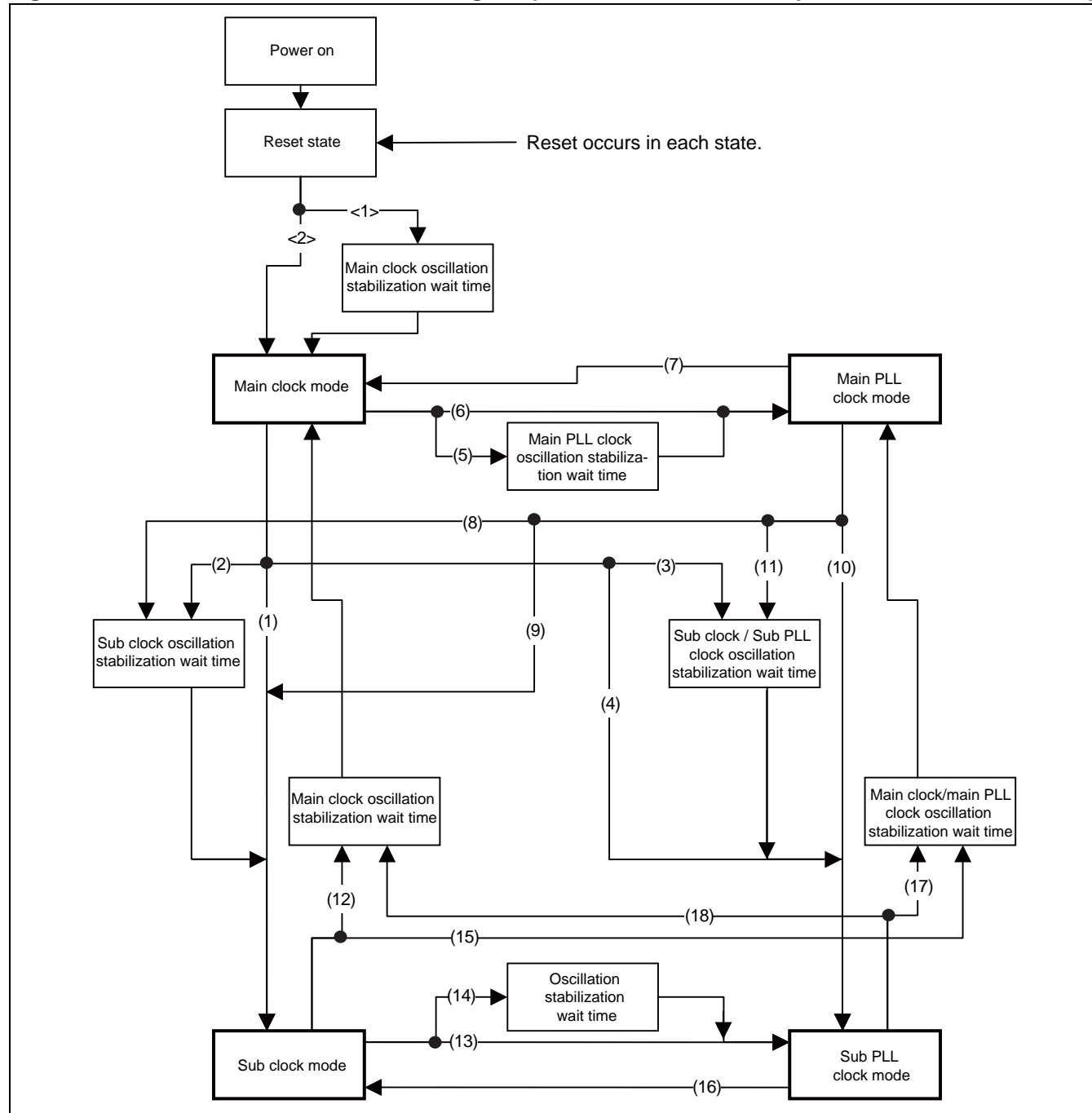
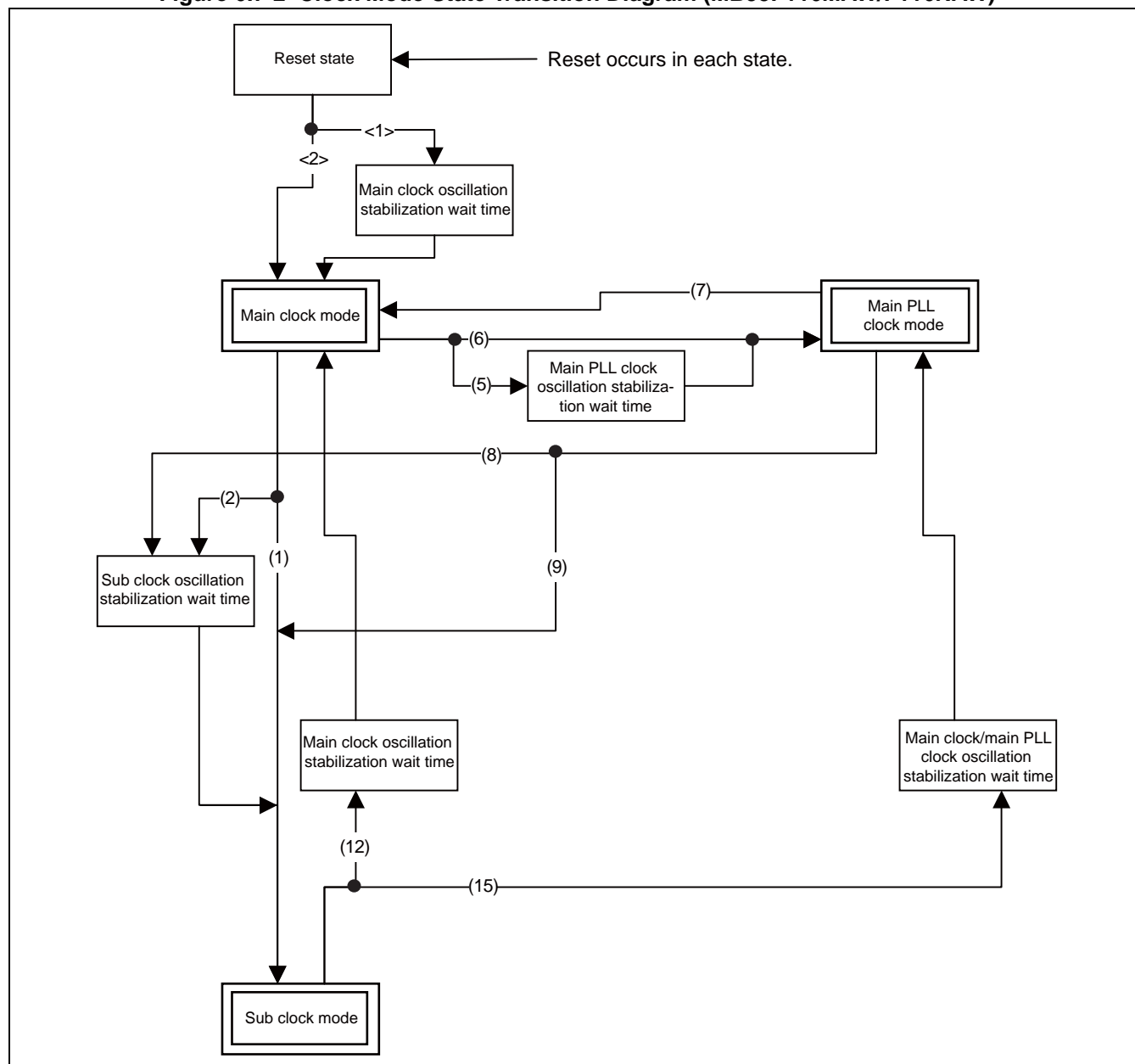


Figure 6.7-2 Clock Mode State Transition Diagram (MB95F116MAW/F116NAW)



**Figure 6.7-3 Clock Mode State Transition Diagram (Single Clock Product)**

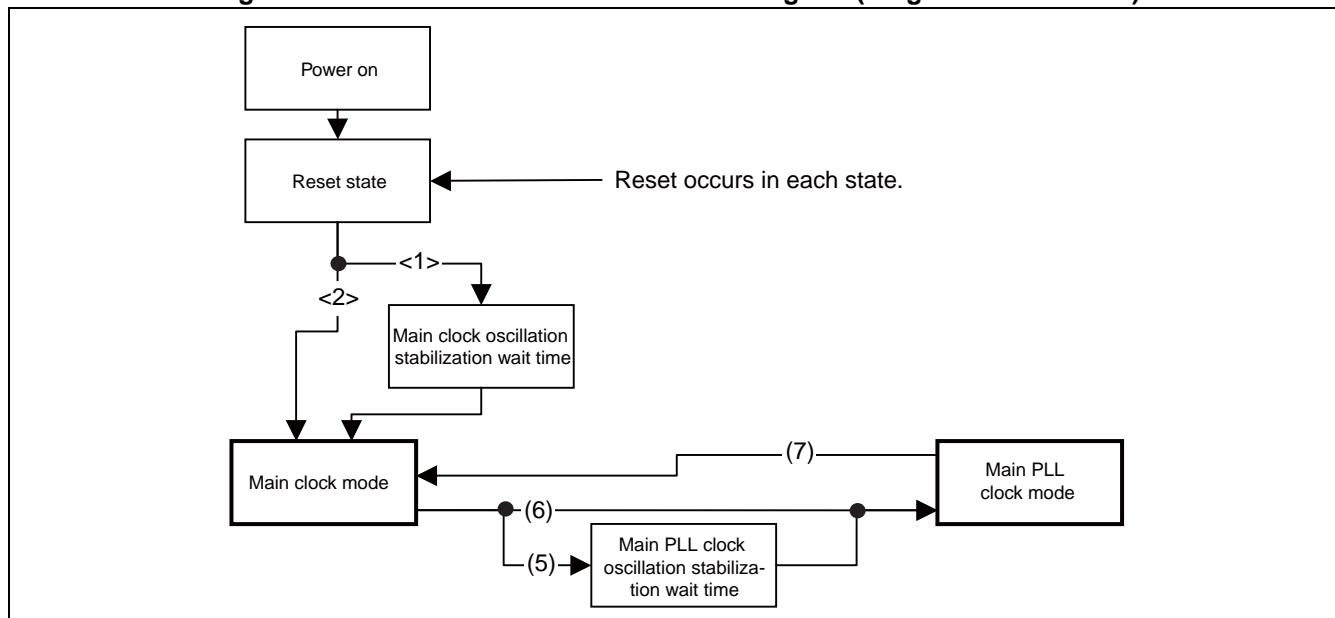


Table 6.7-1 Clock Mode State Transition Table (1 / 2)

	Current State	Next State	Description
<1>	Reset state	Main clock	After a reset, the device waits for the main clock oscillation stabilization wait time to elapse and enters main clock mode. If the reset is a watchdog reset, software reset, or external reset caused in main clock mode or main PLL clock mode, however, the device does not wait for the main clock oscillation stabilization wait time to elapse.
<2>			
(1)	Main clock	Sub clock	The device enters sub clock mode when the system clock selection bits in the system clock control register (SYCC: SCS1, SCS0) are set to "00 <sub>B</sub> ". Note, however, that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on.
(2)			
(3)		Sub PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "01 <sub>B</sub> ", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in main clock mode. Note also that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub PLL clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on. When the device waits for the sub clock oscillation stabilization wait time or sub PLL clock oscillation stabilization wait time, it waits for whichever is longer to elapse.
(4)			
(5)		Main PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "11 <sub>B</sub> ", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the main PLL clock oscillation stabilization wait time to elapse if the main PLL clock has been oscillating according to the setting of the main PLL clock oscillation enable bit in the PLL control register (PLLC: MPEN).
(6)			

Table 6.7-1 Clock Mode State Transition Table (2 / 2)

	Current State	Next State	Description
(7)	Main PLL clock	Main clock	The device enters main clock mode when the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "10 <sub>B</sub> ".
(8)		Sub clock	The device enters sub clock mode when the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "00 <sub>B</sub> ". Note, however, that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main PLL clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on.
(9)			
(10)		Sub PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "01 <sub>B</sub> ", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in main PLL clock mode.
(11)			Note also that the device waits for the sub clock oscillation stabilization wait time to elapse before entering sub PLL clock mode either if the sub clock has been stopped according to the setting of the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS) in main PLL clock mode or if the sub clock oscillation stabilization wait time has not passed immediately after the power is turned on. When the device waits for the sub clock oscillation stabilization wait time or sub PLL clock oscillation stabilization wait time, it waits for whichever is longer to elapse.
(12)	Sub clock	Main clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "10 <sub>B</sub> ", the device enters main clock mode after waiting for the main clock oscillation stabilization wait time.
(13)		Sub PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "01 <sub>B</sub> ", the device enters sub PLL clock mode after waiting for the sub PLL clock oscillation stabilization wait time. Note, however, that the device does not wait for the sub PLL clock oscillation stabilization wait time to elapse if the sub PLL clock has been oscillating according to the setting of the sub PLL clock oscillation enable bit in the PLL control register (PLLC: SPEN) in sub clock mode.
(14)			
(15)		Main PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "11 <sub>B</sub> ", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time or main clock oscillation stabilization wait time to elapse, whichever is longer.
(16)	Sub PLL clock	Sub clock	The device enters sub clock mode when the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "00 <sub>B</sub> ".
(17)		Main PLL clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "11 <sub>B</sub> ", the device enters main PLL clock mode after waiting for the main PLL clock oscillation stabilization wait time or main clock oscillation stabilization wait time to elapse, whichever is longer.
(18)		Main clock	When the system clock selection bits in the system clock control register (SYCC: SCS1, 0) are set to "10 <sub>B</sub> ", the device enters main clock mode after waiting for the main clock oscillation stabilization wait time.

## 6.8 Operations in Low-power Consumption Modes (Standby Modes)

---

**The standby modes available are: sleep mode, stop mode, time-base timer mode, and watch mode.**

---

### ■ Overview of Transitions to and from Standby Mode

The standby modes available are: sleep mode, stop mode, time-base timer mode, and watch mode. The device enters standby mode according to the settings in the standby control register (STBC).

The device is released from standby mode in response to an interrupt or reset. Before transition to normal operation, the device waits for the oscillation stabilization wait time to elapse as required.

When released from standby mode by a reset, the device returns to main clock mode. When released from standby mode by an interrupt, the device enters the clock mode in which the device was before entering the standby mode.

### ■ Pin States in Standby Mode

The pin state setting bit (STBC:SPL) of the standby control register can be used to set the I/O port/peripheral resource pins in the stop mode, time-base timer mode, or watch mode to hold their immediately preceding state or to be placed in a high impedance state.

See "APPENDIX D Pin Status of MB95110B/M series" for the states of all pins in standby modes.



## 6.8.1 Notes on Using Standby Mode

---

Even if the standby control register (STBC) sets standby mode, transition to the standby mode does not take place when an interrupt request has been issued from a peripheral resource. When the device returns from standby mode to the normal operating state in response to an interrupt, the operation that follows varies depending on whether the interrupt request is accepted or not.

---

### ■ Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction.

The device requires four machine clock cycles before entering standby mode after it is set in the standby control register. During that period, the CPU executes the program. To avoid program execution during this transition to standby mode, enter at least three NOP instructions.

The device operates normally if you place instructions other than NOP instructions. In that case, however, note that the device may execute the instructions to be executed after being released from standby mode before entering the standby mode and that the device may enter the standby mode during instruction execution, which is resumed after the device is released from the standby mode (increasing the number of instruction execution cycles).

### ■ Check that Clock-mode Transition has been Completed before Setting Standby Mode.

Before setting standby mode, make sure that clock-mode transition has been completed by comparing the values of the clock mode monitor bit (SYCC: SCM1, 0) and clock mode setting bit (SYCC: SCS1, 0) in the system clock control register.

### ■ An Interrupt Request may Suppress Transition to Standby Mode.

If an attempt is made to set a standby mode while an interrupt request with an interrupt level higher than "11<sub>B</sub>" has been issued, the device ignores the attempt to write to the standby control register and continues instruction execution without entering the standby mode. The device does not enter the standby mode even after having serviced the interrupt.

This behavior is the same as when interrupts are disabled by the interrupt enable flag (CCR:I) and interrupt level bits in the condition code register (CCR:IL 1, 0) of the CPU.

### ■ Standby Mode is Also Canceled when the CPU Rejects Interrupts.

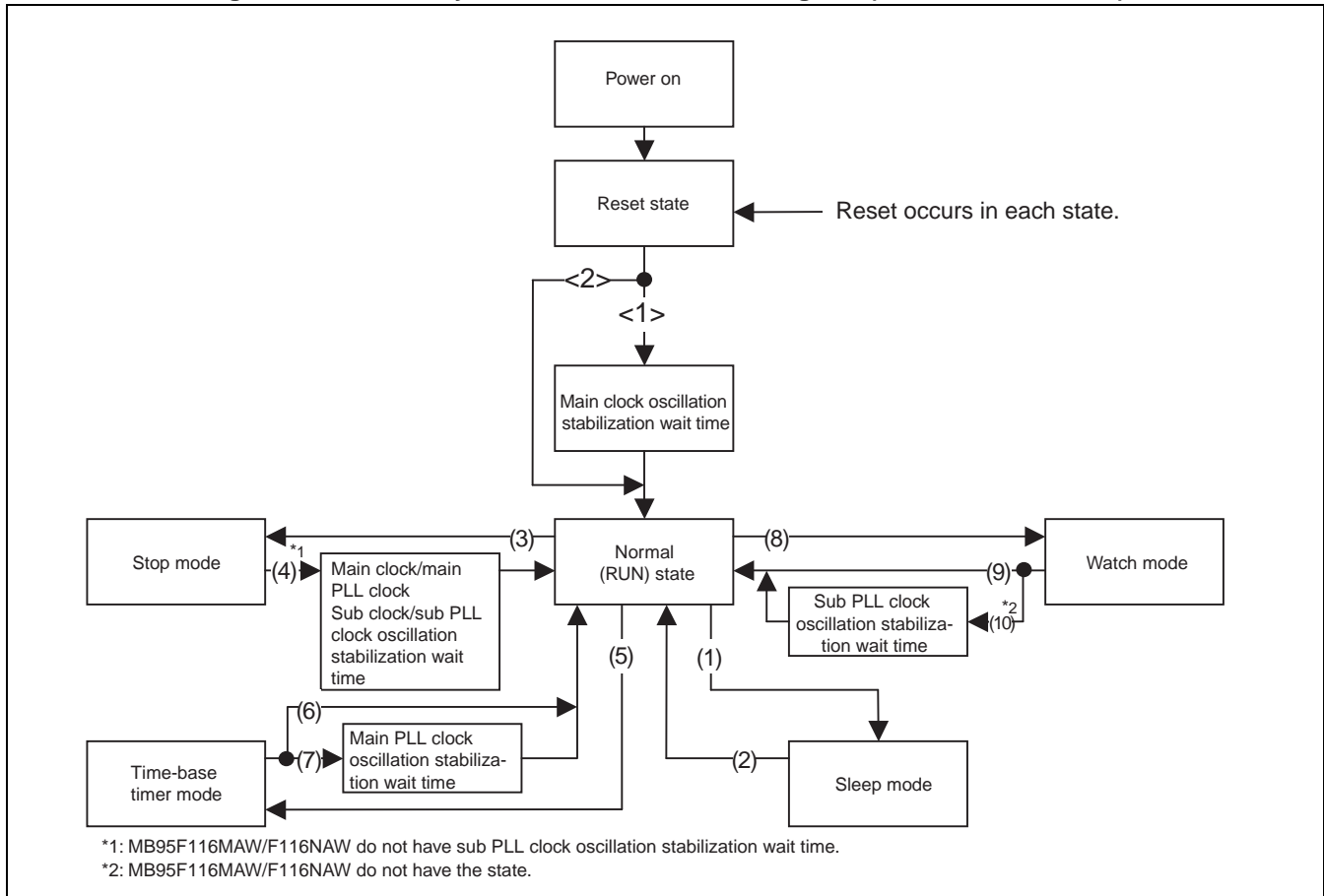
When an interrupt request with an interrupt level higher than "11<sub>B</sub>" is issued in standby mode, the device is released from the standby mode regardless of the settings of the interrupt enable flag (CCR: I) and interrupt level bits (CCR:IL1, 0) of the condition code register of the CPU.

After being released from standby mode, the device services the interrupt when the CPU's condition code register has been set to accept interrupts. If the register has been set to reject interrupts, the device resumes processing from the instruction that follows the last instruction executed before entering the standby mode.

### ■ Standby Mode State Transition Diagram

Figure 6.8-1 and Figure 6.8-2 are standby mode state transition diagrams.

**Figure 6.8-1 Standby Mode State Transition Diagram (Dual Clock Product)**



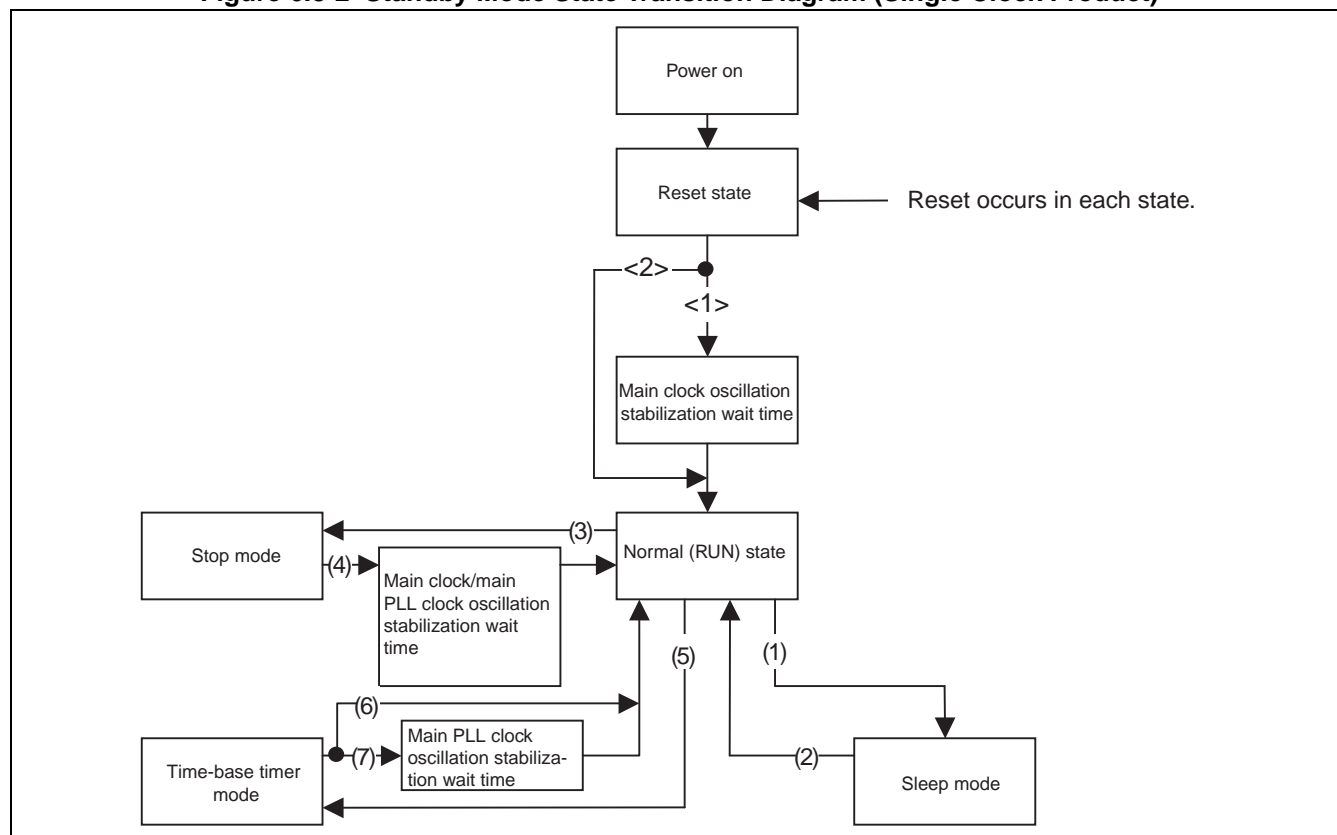
**Figure 6.8-2 Standby Mode State Transition Diagram (Single Clock Product)**

Table 6.8-1 State Transition Diagram (Transitions to and from Standby Modes)

	State Transition	Description
<1>	Normal operation from reset state	After a reset, the device enters main clock mode. If the reset is a power-on reset, the device always waits for the main clock oscillation stabilization wait time to elapse. When the clock mode before the reset is sub clock mode or sub PLL clock mode, the device waits for the main clock oscillation stabilization wait time to elapse. The device waits for it as well when the standby mode is stop mode.
<2>		When the clock mode before the reset is main clock mode or main PLL clock mode and the standby mode is other than stop mode, the device does not wait for the main clock oscillation stabilization wait time to elapse even after entering a reset state in response to a watchdog reset, software reset, or external reset.
(1)	Sleep mode	The device enters sleep mode when "1" is written to the sleep bit in the standby control register (STBC: SLP).
(2)		The device returns to the RUN state in response to an interrupt from a peripheral resource.
(3)	Stop mode	The device enters stop mode when "1" is written to the stop bit in the standby control register (STBC: STP).
(4)		In response to an external interrupt, the device returns to the RUN state after waiting for the oscillation stabilization wait time required for each clock mode. When the device waits for a PLL oscillation stabilization wait time, it waits for the relevant oscillation stabilization wait time or PLL oscillation stabilization wait time to elapse, whichever is longer.
(5)	Time-base timer mode	The device enters time-base timer mode when "1" is written to the watch bit in the standby control register (STBC: TMD) in main clock mode or main PLL clock mode.
(6)		The device returns to the RUN state in response to a time-base timer interrupt, watch prescaler/watch counter interrupt, or external interrupt.
(7)		When the clock mode is main PLL clock mode, the device waits for the main PLL clock oscillation stabilization wait time to elapse. If the main PLL oscillation enable bit in the PLL control register (PLLC: MPEN) contains "1", however, the device does not wait for that time to elapse even when the clock mode is main PLL clock mode.
(8)	Watch mode	The device enters watch mode when "1" is written to the watch bit in the standby control register (STBC: TMD) in sub clock mode or sub PLL clock mode.
(9)		The device returns to the normal operating state in response to a watch prescaler/watch counter interrupt or external interrupt.
(10)		When the clock mode is sub PLL clock mode, the device waits for the sub PLL clock oscillation stabilization wait time to elapse. If the sub PLL oscillation enable bit in the PLL control register (PLLC: SPEN) contains "1", however, the device does not wait for that time to elapse even when the clock mode is sub PLL clock mode.

## 6.8.2 Sleep Mode

---

**Sleep mode stops the operations of the CPU and watchdog timer.**

---

### ■ Operations in Sleep Mode

Sleep mode stops the operating clock for the CPU and watchdog timer. In this mode, the CPU stops while retaining the contents of registers and RAM that exist immediately before the transition to sleep mode, but the peripheral resources except the watchdog timer continue operating.

#### ● Transition to sleep mode

Writing "1" to the sleep bit in the standby control register (STBC:SLP) causes the device to enter sleep mode.

#### ● Cancellation of sleep mode

A reset or an interrupt from a peripheral resource releases the device from sleep mode.

### 6.8.3 Stop Mode

---

**Stop mode stops the main clock.**

---

#### ■ Operations in Stop Mode

Stop mode stops the main clock. In this mode, the device stops all the functions except external interrupt and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to stop mode.

In main clock mode or main PLL clock mode, however, you can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, the watch prescaler and watch counter operate.

##### ● Transition to stop mode

Writing "1" to the stop bit in the standby control register (STBC:STP) causes the device to enter stop mode. At this time, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

In main clock mode or main PLL clock mode, a time-base timer interrupt request may be generated while the device is waiting for main clock oscillation to stabilize after being released from stop mode by an interrupt. If the interrupt interval time of the time-base timer is shorter than the main clock oscillation stabilization wait time, you should disable interrupt requests output from the time-base timer before entering stop mode, thereby preventing unexpected interrupts from occurring.

You should also disable interrupt requests output from the watch prescaler before entering stop mode in sub clock mode or sub PLL clock mode.

##### ● Cancellation of stop mode

The device is released from stop mode in response to a reset or an external interrupt.

In main clock mode or main PLL clock mode, you can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, you can also release the device from stop mode using an interrupt by the watch prescaler or watch counter.

---

**Note:**

When stop mode is canceled via an interrupt, peripheral resources placed into stop mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from stop mode, initialize each peripheral resource as necessary.

---

## 6.8.4 Time-base Timer Mode

---

**Time-base timer mode allows only the main clock oscillation, sub clock oscillation, time-base timer, and watch prescaler to work. The operating clock for the CPU and peripheral resources is stopped in this mode.**

---

### ■ Operations in Time-base Timer Mode

In time-base timer mode, main clock supply is stopped except for the time-base timer. The device stops all the functions except time-base timer, external interrupt and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to time-base timer mode.

You can however start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, the watch prescaler and watch counter operate.

#### ● Transition to time-base timer mode

Writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to enter time-base timer mode if the system clock monitor bits in the system clock control register (SYCC: SCM1, SCM0) are "10<sub>B</sub>" or "11<sub>B</sub>".

The device can enter time-base timer mode only when the clock mode is main clock mode or main PLL clock mode.

Upon transition to time-base timer mode, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

#### ● Cancellation of time-base timer mode

The device is released from time-base timer mode in response to a reset, time-base timer interrupt, or external interrupt.

You can start or stop sub clock oscillation by setting the sub clock oscillation stop bit in the system clock control register (SYCC: SUBS). When the sub clock is oscillating, you can also release the device from time-base timer mode using an interrupt by the watch prescaler or watch counter.

---

#### Note:

When time-base timer mode is canceled via an interrupt, peripheral resources placed into time-base timer mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from time-base timer mode, initialize each peripheral resource as necessary.

---

## 6.8.5 Watch Mode

---

**Watch mode allows only the sub clock and watch prescaler to work. The operating clock for the CPU and peripheral resources is stopped in this mode.**

---

### ■ Operations in Watch Mode

In watch mode, the operating clock for the CPU and peripheral resources is stopped. The device stops all the functions except the watch prescaler, watch counter, external interrupt, and low-voltage detection reset while retaining the contents of registers and RAM that exist immediately before the transition to watch mode.

#### ● Transition to watch mode

Writing "1" to the watch bit in the standby control register (STBC:TMD) causes the device to enter watch mode if the system clock monitor bits in the system clock control register (SYCC: SCM1, SCM0) are "00<sub>B</sub>" or "01<sub>B</sub>".

The device can enter watch mode only when the clock mode is sub clock mode or sub PLL clock mode. Upon transition to watch mode, the states of external pins are retained when the pin state setting bit in the standby control register (STBC:SPL) is "0", and the states of external pins become high impedance when that bit is "1" (those pins are pulled up for which pull-up resistor connection has been selected in the pull-up setting register).

#### ● Cancellation of watch mode

The device is released from watch mode in response to a reset, watch interrupt, or external interrupt.

---

#### Note:

When watch mode is canceled via an interrupt, peripheral resources placed into watch mode during an action resume that action. Therefore, the initial interval time of the interval timer and other similar settings are rendered indeterminate. After recovery from watch mode, initialize each peripheral resource as necessary.

---



## 6.9 Clock Oscillator Circuits

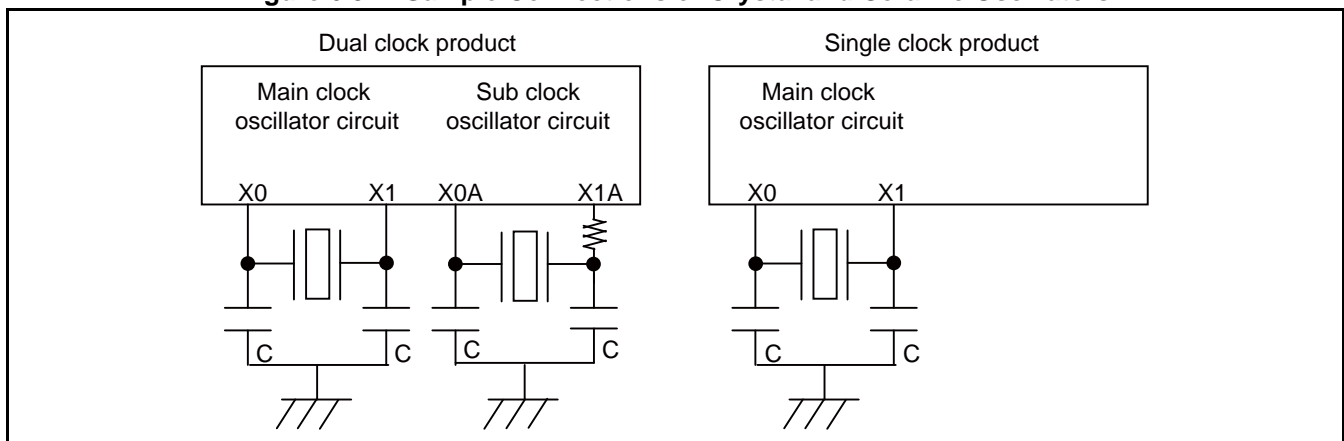
The clock oscillator circuit generates an internal clock with an oscillator connected to or a clock signal input to the clock oscillation pin.

### ■ Clock Oscillator Circuit

#### ● Using crystal and ceramic oscillators

Connect crystal and ceramic oscillators as shown in Figure 6.9-1.

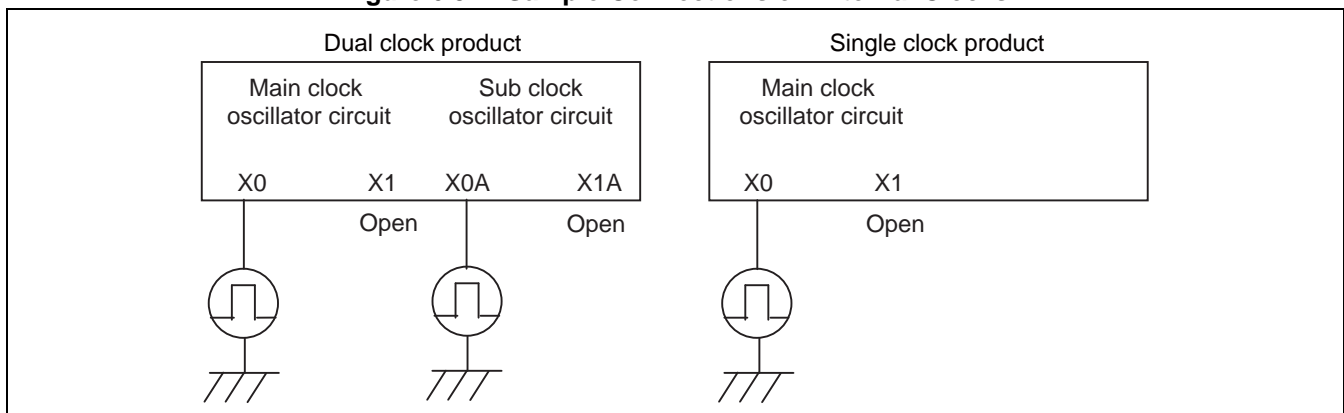
**Figure 6.9-1 Sample Connections of Crystal and Ceramic Oscillators**



#### ● Using external clock

As shown in Figure 6.9-2, connect the external clock to the X0 pin while leaving the X1 pin open. To supply the sub clock from an external source, connect the external clock to the X0A pin while leaving the X1A pin open.

**Figure 6.9-2 Sample Connections of External Clocks**



---

Note:

If you use only the main clock without using sub clock oscillation on a dual clock product and it enters sub clock mode for some reason, there is no solution to recovering its operation as there is no clock supply available. If you use the main clock alone, therefore, be sure to select a single clock product.

---

## 6.10 Overview of Prescaler

---

**The prescaler generates the count clock source for various peripheral resources from the machine clock (MCLK) and the count clock output from the time-base timer.**

---

### ■ Prescaler

The prescaler generates the count clock source for various peripheral resources from the machine clock (MCLK) that drives the CPU and the count clock ( $2^7/F_{CH}$  and  $2^8/F_{CH}$ ) output from of the time-base timer. The count clock source is a clock frequency-divided by the prescaler or a buffered clock, used by the peripheral resources listed below.

Note that the prescaler has no control register and operates continuously driven by the machine clock (MCLK) and the count clock ( $2^7/F_{CH}$  and  $2^8/F_{CH}$ ) of the time-base timer.

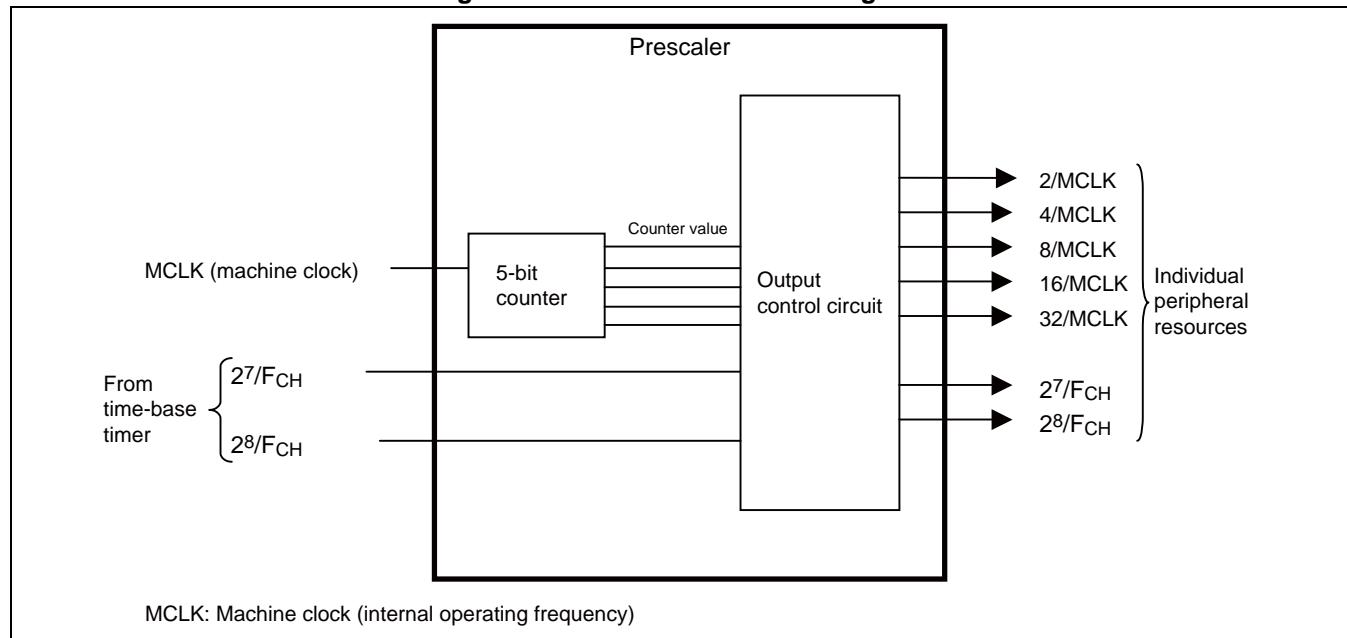
- 8/16-bit compound timer 0, 1
- 8/16-bit PPG timer 0, 1
- 16-bit PPG timer 0
- UART/SIO baud rate generator 0
- 8/10-bit A/D converter

## 6.11 Configuration of Prescaler

Figure 6.11-1 is a block diagram of the prescaler.

### ■ Prescaler Block Diagram

Figure 6.11-1 Prescaler Block Diagram



- 5-bit counter

The machine clock (MCLK) is counted by a 5-bit counter and the count value is output to the output control circuit.

- Output control circuit

Based on the 5-bit counter value, this circuit supplies clocks generated by frequency-dividing the machine clock (MCLK) by 2, 4, 8, 16, or 32 to individual peripheral resources. The circuit also buffers the clock from the time-base timer ( $2^7/F_{CH}$  and  $2^8/F_{CH}$ ) and supplies it to the peripheral resources.

### ■ Input Clock

The prescaler uses the machine clock or the clock output from the time-base timer as the input clock.

### ■ Output Clock

The prescaler supplies clocks to the 8/10-bit compound timer, 8/16-bit PPG timer, 16-bit PPG timer, UART/SIO dedicated baud rate generator, and 8/10-bit A/D converter.

## 6.12 Operating Explanation of Prescaler

The prescaler generates count clock sources to individual peripheral resources.

### ■ Operations of Prescaler

The prescaler generates count clock sources from the frequency-divided version of the machine clock (MCLK) and buffered signals from the time-base timer ( $2^7/F_{CH}$ ,  $2^8/F_{CH}$ ) and supplies them to individual peripheral resources. The prescaler remains operating as long as the machine clock and time-base timer clocks are supplied.

Table 6.12-1 lists the count clock sources generated by the prescaler.

**Table 6.12-1 Count Clock Sources Generated by Prescaler**

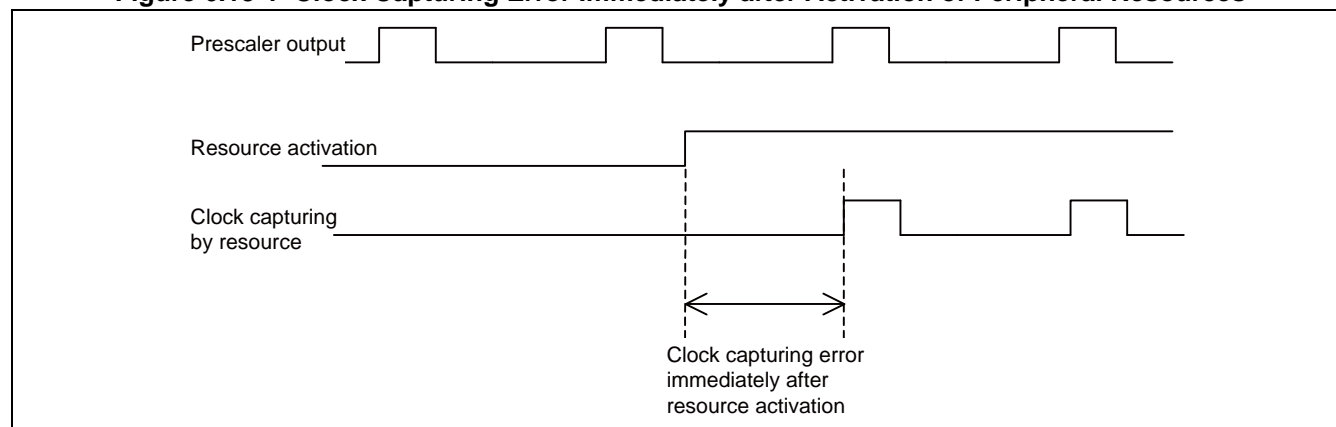
Count Clock Source Cycle	Cycle ( $F_{CH}=10\text{MHz}$ , MCLK=10MHz)		Cycle ( $F_{CH}=16\text{MHz}$ , MCLK=16MHz)		Cycle ( $F_{CH}=16.25\text{MHz}$ , MCLK=16.25MHz)	
2/MCLK	MCLK/2	(5MHz)	MCLK/2	(8MHz)	MCLK/2	(8.125MHz)
4/MCLK	MCLK/4	(2.5MHz)	MCLK/4	(4MHz)	MCLK/4	(4.0625MHz)
8/MCLK	MCLK/8	(1.25MHz)	MCLK/8	(2MHz)	MCLK/8	(2.0313MHz)
16/MCLK	MCLK/16	(0.625MHz)	MCLK/16	(1MHz)	MCLK/16	(1.0156MHz)
32/MCLK	MCLK/32	(0.3125MHz)	MCLK/32	(0.5MHz)	MCLK/32	(0.5078MHz)
$2^7/F_{CH}$	$F_{CH}/2^7$	(78kHz)	$F_{CH}/2^7$	(125kHz)	$F_{CH}/2^7$	(127kHz)
$2^8/F_{CH}$	$F_{CH}/2^8$	(39kHz)	$F_{CH}/2^8$	(62.5kHz)	$F_{CH}/2^8$	(63.5kHz)

## 6.13 Notes on Use of Prescaler

This section gives notes on using the prescaler.

The prescaler uses the machine clock and time-base timer clock and operates continuously while these clocks are running. Accordingly, the operations of individual peripheral resources immediately after they are activated may involve an error of up to one cycle of the clock source captured by the resource, depending on the prescaler output value.

**Figure 6.13-1 Clock Capturing Error Immediately after Activation of Peripheral Resources**



The prescaler count value affects the following resources:

- UART/SIO
- 8/16-bit compound timer
- 8/16-bit PPG
- 16-bit PPG
- 8/10-bit A/D converter



# ***CHAPTER 7***

---

# ***RESET***

**This section describes the reset operation.**

7.1 Reset Operation

7.2 Reset Source Register (RSRR)

7.3 Notes on Using Reset



## 7.1 Reset Operation

When a reset factor occurs, the CPU stops the current execution immediately and enters the reset release wait state. When the device is released from the reset, the CPU reads mode data and the reset vector from internal ROM (mode fetch). When the power is turned on or when the device is released from a reset in sub clock mode, sub-PLL clock mode, or stop mode, the CPU performs mode fetch after the oscillation stabilization wait time has passed.

### ■ Reset Factors

Resets are classified into five reset factors.

**Table 7.1-1 Reset Sources**

Reset Sources	Reset Condition
External reset	"L" level input to the external reset pin
Software reset	"1" is written to the software reset bit (STBC: SRST) in the standby control register.
Watchdog reset	The watchdog timer causes an overflow.
Power-on reset/ low-voltage detection reset	The power is turned on or the supply voltage falls below the detected voltage. (Option)
Clock supervisor reset	Abnormal Stop of Clock Oscillation (Option)

#### ● External reset

An external reset is generated upon "L" level input to the external reset pin ( $\overline{\text{RST}}$ ).

An externally input reset signal is accepted asynchronously via the internal noise filter and generates an internal reset signal in synchronization with the machine clock to initialize the internal circuit. Consequently, a clock is necessary for internal circuit initialization. Clock input is therefore necessary for operation with an external clock. Note, however, that external pins (including I/O ports and peripheral resources) are reset asynchronously. Additionally, there are standard pulse-width values for external reset input. If the value is below the standard, the reset may not be accepted.

The standard value is listed on the data sheet. Please design your external reset circuit so that this standard is met.

#### ● Software reset

Writing "1" to the software reset bit of the standby control register (STBC:SRST) generates a software reset.

#### ● Watchdog reset

After the watchdog timer starts, a watchdog reset is generated if the watchdog timer is not cleared within a preset amount of time.

**● Power-on reset/low-voltage detection reset (Option)**

A power-on reset is generated when the power is turned on.

Some 5-V products have a low-voltage detection reset circuit (option) integrated.

The low-voltage detection reset circuit generates a reset if the power supply voltage falls below a predetermined level.

The logical function of the low-voltage detection reset is completely equivalent to the power-on reset. All the text in this manual concerning power-on resets applies to low-voltage detection resets as well.

For details about low-voltage detection resets, see "CHAPTER 25 LOW-VOLTAGE DETECTION RESET CIRCUIT".

**● Clock Supervisor Reset (Option)**

Some 5V products have the (optional) clock supervisor.

The clock supervisor monitors the main and sub clocks and generates a reset when the oscillation stops due to not given state transition but any abnormality. After reset, a clock occurred in the built-in RC oscillation circuit is provided internally.

For details on the clock supervisor, see "CHAPTER 26 CLOCK SUPERVISOR".

**■ Reset Time**

In the case of a software reset or watchdog reset, the reset time consists of a total of three machine clock cycles: one machine clock cycle at the machine clock frequency selected before the reset, and two machine clock cycles at the machine clock frequency initially set after the reset (1/32 of the main clock frequency). However, the reset time may be extended in machine clock cycles of the frequency selected before the reset, via the RAM access protection function which suppresses resets during RAM access. In addition, when in main clock oscillation stabilization standby mode, the reset time is further extended for the oscillation stabilization wait time.

External resets and resets are also affected by the RAM access protection function and main clock oscillation stabilization wait time.

In the case of a power-on reset or low-voltage detection reset, the reset continues during the oscillation stabilization wait time.

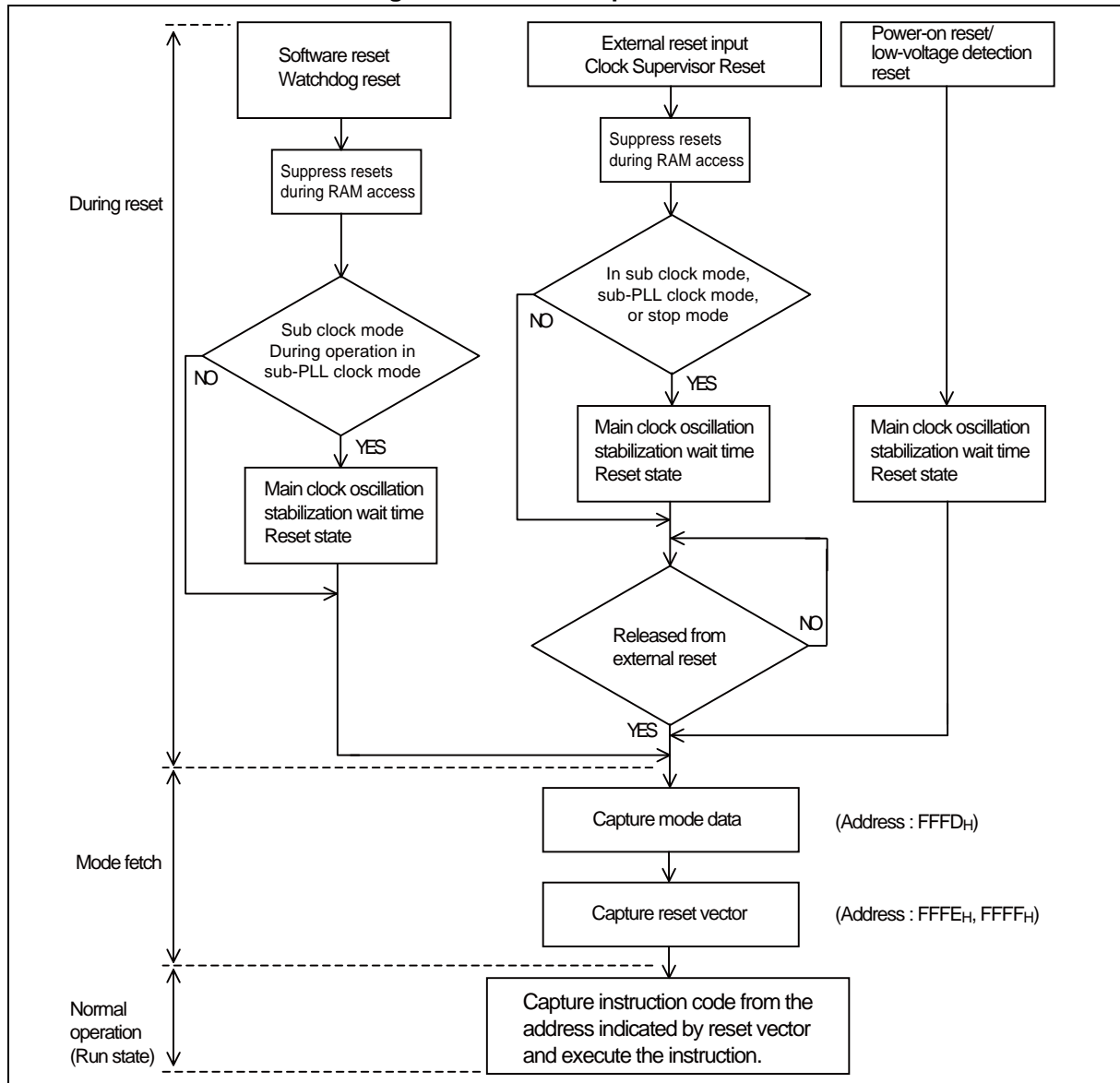
**■ Reset Output**

The  $\overline{\text{RST}}$  pin of 5 V products with the reset (For details, see Table 1.2-1.) outputs "L" level during reset time. However, a reset pin does not output "L" level in the case of an external reset.

The  $\overline{\text{RST}}$  pin of 3 V products and 5 V products without the reset outputs do not have an output function.

## ■ Overview of Reset Operation

Figure 7.1-1 Reset Operation Flow



In the case of a power-on reset/low-voltage detection reset, and a reset when in sub clock mode, sub-PLL clock mode, or stop mode, the CPU performs mode fetch after the main clock oscillation stabilization wait time has elapsed. If the external reset input is not cleared after the oscillation stabilization wait time has elapsed, the CPU performs mode fetch after the external reset input is cleared.

## ■ Effect of Reset on RAM Contents

When a reset occurs, the CPU halts the operation of the command currently being executed, and enters the reset status. During RAM access execution, however, RAM access protection causes an internal reset signal to be generated in synchronization with the machine clock, after RAM access has ended. This function prevents a word-data write operation from being cut off by a reset after one byte.

#### ■ Pin State During a Reset

When a reset occurs, all of the I/O ports and peripheral resource pins remain in a high impedance state until setup is performed by software after the reset is released.

---

#### Note:

Connect a pull-up resistor to those pins which remain at high impedance during a reset to prevent the devices the pins from malfunctioning.

---

See "APPENDIX D Pin Status of MB95110B/M series" for details about the states of all pins during a reset.

## 7.2 Reset Source Register (RSRR)

The reset source register indicates the source or factor causing a reset that has been generated.

### ■ Configuration of Reset Source Register (RSRR)

Figure 7.2-1 Reset Source Register (RSRR)

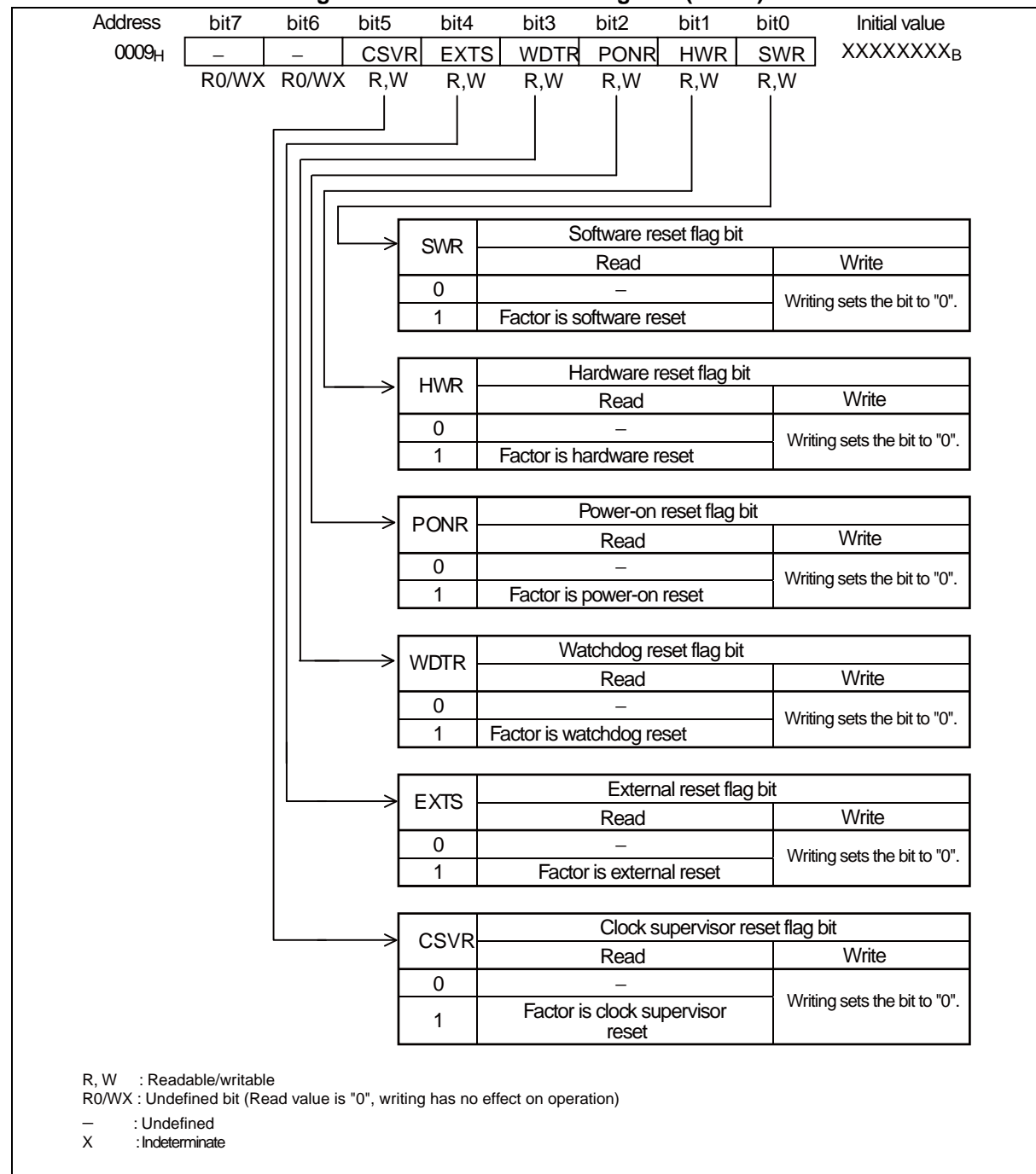


Table 7.2-1 Functions of Bits in Reset Source Register (RSRR)

Bit name		Function
bit7, bit6	Undefined bits	The read value is always "0". These bits are read-only. Writing has no effect on operation.
bit5	CSVR: Clock supervisor reset flag bit	This bit is set to "1" to indicate that a clock supervisor reset has occurred. Otherwise, the bit retains the value existing before the clock supervisor reset occurred. <ul style="list-style-type: none"> <li>Read or write access (0 or 1) to this bit sets it to "0".</li> <li>The bit value is always "0" in product types that do not have the clock supervisor function.</li> </ul> Writing has no effect on the operation.
bit4	EXTS: External reset flag bit	This bit is set to "1" to indicate that an external reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. <ul style="list-style-type: none"> <li>Read or write access (0 or 1) to this bit sets it to "0".</li> </ul>
bit3	WDTR: watchdog reset flag bit	This bit is set to "1" to indicate that an watchdog reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. <ul style="list-style-type: none"> <li>Read or write access (0 or 1) to this bit sets it to "0".</li> </ul>
bit2	PONR: Power-on reset flag bit	This bit is set to "1" to indicate that a power-on reset or low-voltage detection reset (option) has occurred. Otherwise, the bit retains the value existing before the reset occurred. <ul style="list-style-type: none"> <li>The low-voltage detection reset function is provided for specific models.</li> <li>Read or write access (0 or 1) to this bit sets it to "0".</li> </ul>
bit1	HWR: Hardware reset flag bit	This bit is set to "1" to indicate that a reset other than a software reset has occurred. When any of bits 2 to 5 is set to "1", therefore, this bit is set to "1" as well. Otherwise, the bit retains the value existing before the reset occurred. <ul style="list-style-type: none"> <li>Read or write access (0 or 1) to this bit sets it to "0".</li> </ul>
bit0	SWR: Software reset flag bit	This bit is set to "1" to indicate that a software reset has occurred. Otherwise, the bit retains the value existing before the reset occurred. <ul style="list-style-type: none"> <li>Read or write access (0 or 1) to this bit or a power-on reset sets it to "0".</li> </ul>

**Note:**

Reading the reset source register clears its contents. To use the reset source register for calculation, therefore, you should move the contents of the register to RAM in advance.

■ **Status of Reset Source Register (RSRR)**

**Table 7.2-2 Status of Reset Source Register**

<b>Reset Sources</b>	—	—	<b>CSVR</b>	<b>EXTS</b>	<b>WDTR</b>	<b>PONR</b>	<b>HWR</b>	<b>SWR</b>
Power-on reset/ low-voltage detection reset	—	—	×	×	×	1	1	0
Software reset	—	—	Δ	Δ	Δ	Δ	Δ	1
Watchdog reset	—	—	Δ	Δ	1	Δ	1	Δ
External reset	—	—	Δ	1	Δ	Δ	1	Δ
Clock supervisor reset	—	—	1	Δ	Δ	Δ	1	Δ

1 : Flag set

Δ : Previous state saved

×

CSVR : This bit is set to "1" to indicate that a clock supervisor reset has occurred  
(Always "0" if there is no clock supervisor option)

EXTS : This bit is set to "1" to indicate that an external reset has occurred.

WDTR : This bit is set to "1" to indicate that a watchdog reset has occurred.

PONR : This bit is set to "1" to indicate that a power-on reset or low-voltage detection reset (option) has occurred.

HWR : The bit value "1" indicates that a reset source occurs from either CSVR, EXTS, WDTR, or PONR.

SWR : This bit is set to "1" to indicate that a software reset has occurred.

## 7.3 Notes on Using Reset

---

This section explains the notes on using Reset.

---

### ■ Notes on Using Reset

#### ● Initialization of the main clock stop detection bit of clock supervisor

The main clock stop detection bit (CSVCR:MM) of clock supervisor is initialized only by power-on reset and external reset.

The bit is not initialized by the watchdog timer reset/software reset/clock supervisor reset. Therefore, if one of these resets is issued, the CR clock mode continues.

#### ● Initialization of register and bit by reset source

Some registers and bits are not initialized by reset source.

For the reset source register (RSRR), which of the bit is initialized depends on the reset source.

- The main clock stop detection bit (CSVCR:MM) of clock supervisor is initialized only by power-on reset and external reset.
- The CR oscillation enable bit (CSVCR:RCE) of clock supervisor is initialized only by power-on reset/ external reset.
- The main clock monitoring enable bit (CSVCR:MSVE) of clock supervisor is initialized only by power-on reset.
- The oscillation stabilization wait time setting register (WATR) of clock control block is initialized only by power-on reset.





# ***CHAPTER 8***

---

# ***INTERRUPTS***

**This section explains the interrupts.**

## 8.1 Interrupts

## 8.1 Interrupts

---

**This section explains the interrupts.**

---

### ■ Overview of Interrupts

The F<sup>2</sup>MC-8FX family has 24 interrupt request input lines corresponding to peripheral resources, for each of which an interrupt level can be set independently.

When a peripheral resource generates an interrupt request, the interrupt request is output to the interrupt controller. The interrupt controller checks the interrupt level of that interrupt request and then passes the occurrence of the interrupt to the CPU. The CPU services the interrupt according to the interrupt acceptance status. Interrupt requests also release the device from standby mode to resume instruction execution.

### ■ Interrupt Requests from Peripheral Resources

Table 8.1-1 shows the interrupt requests corresponding to the peripheral resources. When an interrupt is accepted, a branch to the interrupt service routine takes place with the content of the interrupt vector table address corresponding to the interrupt request as the address of the branch destination.

The priority for each interrupt request can be set to one of four levels using the interrupt level setting registers (ILR0 to ILR5).

If another interrupt request with the same or lower level occurs during execution of the interrupt service routine, the interrupt is processed after the current interrupt handler routine completes. If interrupt requests of the same level occur at the same time, IRQ0 is assigned the highest priority.

Table 8.1-1 Interrupt Requests and Interrupt Vectors

Interrupt request	Vector table address		Bit name of interrupt level setting register	Priority for equal-level Interrupt requests (generated simultaneously)
	Upper	Lower		
(Reset vector)	FFFE <sub>H</sub>	FFFF <sub>H</sub>	-	<div>High</div> <div>↑</div> <div>↓</div> <div>Low</div>
(Mode data)	-	FFFD <sub>H</sub>	-	
IRQ0	FFFA <sub>H</sub>	FFFB <sub>H</sub>	L00 [1:0]	
IRQ1	FFF8 <sub>H</sub>	FFF9 <sub>H</sub>	L01 [1:0]	
IRQ2	FFF6 <sub>H</sub>	FFF7 <sub>H</sub>	L02 [1:0]	
IRQ3	FFF4 <sub>H</sub>	FFF5 <sub>H</sub>	L03 [1:0]	
IRQ4	FFF2 <sub>H</sub>	FFF3 <sub>H</sub>	L04 [1:0]	
IRQ5	FFF0 <sub>H</sub>	FFF1 <sub>H</sub>	L05 [1:0]	
IRQ6	FFEE <sub>H</sub>	FFEF <sub>H</sub>	L06 [1:0]	
IRQ7	FFEC <sub>H</sub>	FFED <sub>H</sub>	L07 [1:0]	
IRQ8	FFEA <sub>H</sub>	FFEB <sub>H</sub>	L08 [1:0]	
IRQ9	FFE8 <sub>H</sub>	FFE9 <sub>H</sub>	L09 [1:0]	
IRQ10	FFE6 <sub>H</sub>	FFE7 <sub>H</sub>	L10 [1:0]	
IRQ11	FFE4 <sub>H</sub>	FFE5 <sub>H</sub>	L11 [1:0]	
IRQ12	FFE2 <sub>H</sub>	FFE3 <sub>H</sub>	L12 [1:0]	
IRQ13	FFE0 <sub>H</sub>	FFE1 <sub>H</sub>	L13 [1:0]	
IRQ14	FFDE <sub>H</sub>	FFDF <sub>H</sub>	L14 [1:0]	
IRQ15	FFDC <sub>H</sub>	FFDD <sub>H</sub>	L15 [1:0]	
IRQ16	FFDA <sub>H</sub>	FFDB <sub>H</sub>	L16 [1:0]	
IRQ17	FFD8 <sub>H</sub>	FFD9 <sub>H</sub>	L17 [1:0]	
IRQ18	FFD6 <sub>H</sub>	FFD7 <sub>H</sub>	L18 [1:0]	
IRQ19	FFD4 <sub>H</sub>	FFD5 <sub>H</sub>	L19 [1:0]	
IRQ20	FFD2 <sub>H</sub>	FFD3 <sub>H</sub>	L20 [1:0]	
IRQ21	FFD0 <sub>H</sub>	FFD1 <sub>H</sub>	L21 [1:0]	
IRQ22	FFCE <sub>H</sub>	FFCF <sub>H</sub>	L22 [1:0]	
IRQ23	FFCC <sub>H</sub>	FFCD <sub>H</sub>	L23 [1:0]	

For interrupt sources, see "APPENDIX B Table of Interrupt Causes".

## 8.1.1 Interrupt Level Setting Registers (ILR0 to ILR5)

The interrupt level setting registers (ILR0 to ILR5) contain 24 pairs of bits assigned for the interrupt requests from different peripheral resources. Each pair of bits (interrupt level setting bits as two-bit data) sets each interrupt level.

### ■ Configuration of Interrupt Level Setting Registers (ILR0 to ILR5)

Figure 8.1-1 Configuration of Interrupt Level Setting Registers

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ILR0	00079 <sub>H</sub>	L03	[1:0]	L02	[1:0]	L01	[1:0]	L00	[1:0]	R/W 11111111 <sub>B</sub>
ILR1	0007A <sub>H</sub>	L07	[1:0]	L06	[1:0]	L05	[1:0]	L04	[1:0]	R/W 11111111 <sub>B</sub>
ILR2	0007B <sub>H</sub>	L11	[1:0]	L10	[1:0]	L09	[1:0]	L08	[1:0]	R/W 11111111 <sub>B</sub>
ILR3	0007C <sub>H</sub>	L15	[1:0]	L14	[1:0]	L13	[1:0]	L12	[1:0]	R/W 11111111 <sub>B</sub>
ILR4	0007D <sub>H</sub>	L19	[1:0]	L18	[1:0]	L17	[1:0]	L16	[1:0]	R/W 11111111 <sub>B</sub>
ILR5	0007E <sub>H</sub>	L23	[1:0]	L22	[1:0]	L21	[1:0]	L20	[1:0]	R/W 11111111 <sub>B</sub>

The interrupt level setting registers assign each pair of bits for a different interrupt request. The values of interrupt level setting bits in these registers specify interrupt service priorities (interrupt levels 0 to 3).

The interrupt level setting bits are compared with the interrupt level bits in the condition code register (CCR: IL1, IL0).

When interrupt level 3 is set for an interrupt request, the CPU ignores the interrupt request.

Table 8.1-2 shows the relationships between interrupt level setting bits and interrupt levels.

Table 8.1-2 Relationships Between Interrupt Level Setting Bits and Interrupt Levels

LXX[1:0]	Interrupt Level	Priority
00	0	<div style="text-align: center;"> High  ↑  ↓  Low (No interrupt) </div>
01	1	
10	2	
11	3	

XX:00 to 23 Corresponding interrupt number

During execution of a main program, the interrupt level bits in the condition code register (CCR: IL1, IL0) are usually "11<sub>B</sub>".

**MB95110B/M Series****8.1.2 Interrupt Processing Steps**

---

**When an interrupt request is generated by a peripheral resource, the interrupt controller passes the interrupt level to the CPU. When the CPU is ready to accept interrupts, it temporarily halts the program currently being executed and executes an interrupt service routine.**

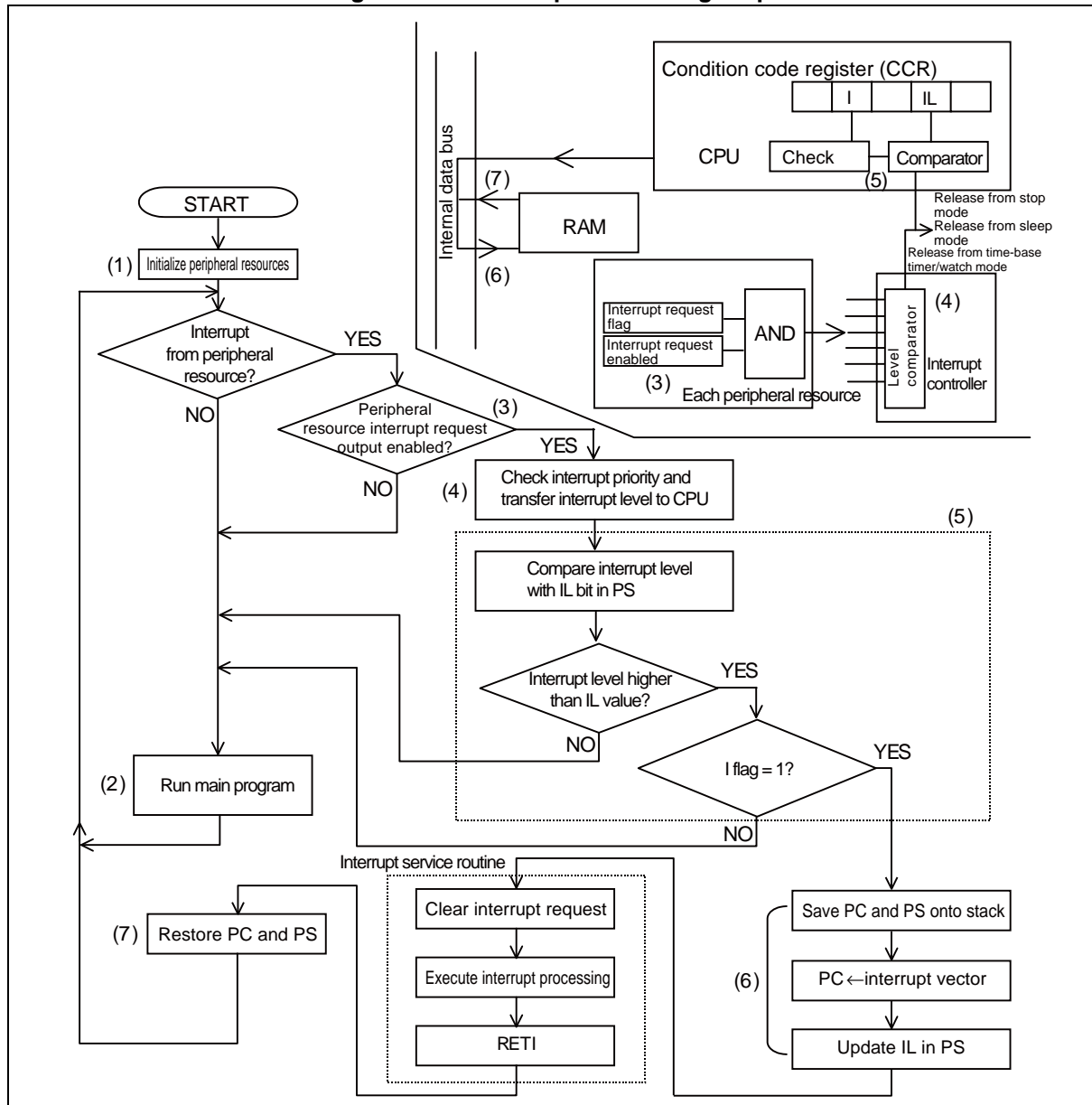
---

**■ Interrupt Processing**

The procedure of processing an interrupt takes the following steps: the generation of an interrupt resource in a peripheral resource, the execution of the main program, the setting of the interrupt request flag bit, the evaluation of the interrupt request enable bit, the evaluation of interrupt level (ILR0 to ILR5 and CCR:IL1, IL0), the checking for any equal-level interrupt request, and the evaluation of the interrupt enable flag (CCR:I).

Figure 8.1-2 illustrates the steps to take for interrupt processing.

Figure 8.1-2 Interrupt Processing Steps



- (1) Any interrupt request is disabled immediately after a reset. In the peripheral resource initialization program, initialize those peripheral resources which generate interrupts and set their interrupt levels in their respective interrupt level setting registers (ILR0 to ILR5) before starting operating the peripheral resources. The interrupt level can be set to 0, 1, 2, or 3. Level 0 is given the highest priority, and level 1 the second highest. Setting level 3 for a peripheral resource disables interrupts from that resource.
- (2) Execute the main program (or the interrupt processing routine for nested interrupts).
- (3) When an interrupt is triggered in a peripheral resource, the interrupt request flag bit of the peripheral resource is set to "1". If the interrupt request enable bit of the peripheral resource has been set to enable interrupts, the interrupt request is then output to the interrupt controller.
- (4) The interrupt controller always monitors interrupt requests from individual peripheral resources and transfers the highest-priority interrupt level to the CPU among the interrupt levels of the currently generated interrupt requests. The relative priority to be assigned if another request with the same interrupt level occurs simultaneously is also determined at this time.
- (5) If the received interrupt level or priority is lower than the level set in the interrupt level bits in the condition code register (CCR: IL1, IL0), the CPU checks the content of the interrupt enable flag (CCR:I) and, if interrupts are enabled (CCR:I = 1), accepts the interrupt.
- (6) The CPU pushes the contents of the program counter (PC) and program status (PS) register onto the stack, fetches the start address of the interrupt processing routine from the corresponding interrupt vector table, changes the value of the interrupt level bits in the condition code register (CCR: IL1, IL0) to the value of the received interrupt level, then starts the execution of the interrupt processing routine.
- (7) Finally, the CPU uses the RETI instruction to restore the program counter (PC) and program status (PS) values from the stack and resumes execution from the instruction that follows the instruction executed prior to the interrupt.

---

**Note:**

The interrupt request flag bits of peripheral resources are not automatically cleared to "0" after an interrupt request is accepted. The bits must therefore be cleared to "0" by a program (by writing "0" to the interrupt request flag bit) in the interrupt processing routine.

---

An interrupt causes the device to recover from standby mode (low power consumption mode). For details, see Section "6.8 Operations in Low-power Consumption Modes (Standby Modes)".



### 8.1.3 Nested Interrupts

You can set different interrupt levels for two or more interrupt requests from peripheral resources in the interrupt level setting registers (ILR0 to ILR5) to process the nested interrupts.

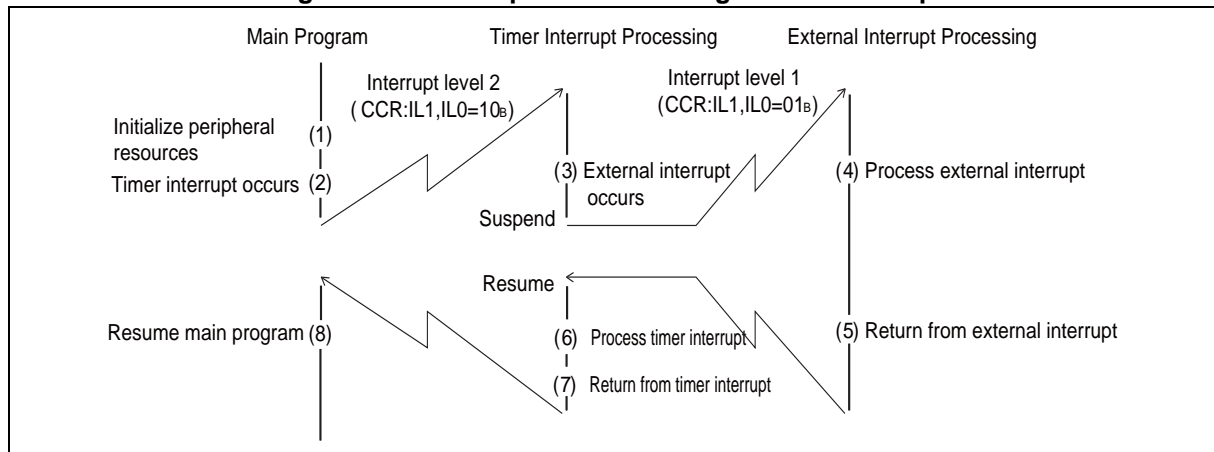
#### ■ Nested Interrupts

If an interrupt request of higher-priority interrupt level occurs while an interrupt service routine is being executed, the CPU halts processing of the current interrupt and accepts the higher-priority interrupt request. The interrupt level can be set to 0 to 3. If it is set to 3, the CPU will accept no interrupt request.

[Example: Nested interrupts]

To assign higher priority to external interrupts over timer interrupts as an example of processing nested-interrupts, set the timer interrupt and external interrupt levels to 2 and 1, respectively. If an external interrupt occurs while a timer interrupt is being processed with these settings in use, the interrupts are processed as shown in Figure 8.1-3.

**Figure 8.1-3 Example of Processing Nested Interrupts**



- While a timer interrupt is being processed, the interrupt level bits in the condition code register (CCR: IL1, IL0) hold the same value as that of the interrupt level setting registers (ILR0 to ILR5) corresponding to the current timer interrupt (level 2 in this example). If an interrupt request with a higher-priority interrupt level (level 1 in the example) occurs, the higher-priority interrupt is processed preferentially.
- To temporarily disable nested interrupt processing while a timer interrupt is being processed, set the interrupt enable flag in the condition code register to disable interrupts (CCR:I = 0) or set the interrupt level bits (CCR: IL1, IL0) to "00<sub>B</sub>".
- Executing the interrupt return instruction (RETI) after interrupt processing is completed restores the program counter (PC) and program status (PS) values saved in a stack and resumes the processing of the interrupted program. Restoring the program status (PS) also restores the condition code register (CCR) to its value existing prior to the interrupt.

**MB95110B/M Series****8.1.4 Interrupt Processing Time**

The time between an interrupt request being generated and control being passed to the interrupt processing routine is equal to the sum of the time until the currently executing instruction completes and the interrupt handling time (time required to initiate interrupt processing). This time consists of a maximum of 26 machine clock cycles.

**■ Interrupt Processing Time**

The interrupt request sampling wait time and interrupt handling time intervene between the occurrence and acceptance of an interrupt request and the execution of the relevant interrupt service routine.

● **Interrupt request sampling wait time**

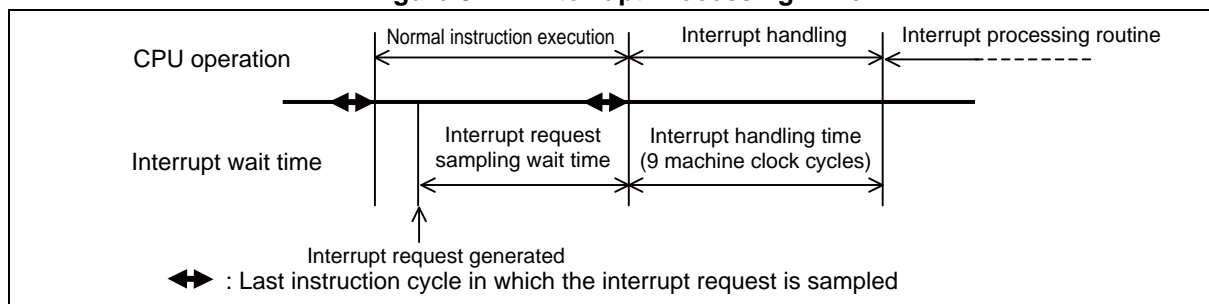
Whether an interrupt request has occurred is determined through the sampling of the interrupt request during the last cycle of each instruction. The CPU cannot therefore recognize interrupt requests during the execution of each instruction. The maximum length of this delay occurs if the interrupt request is generated immediately after the DIVU instruction requiring the longest instruction cycle (17 machine clock cycles) starts executing.

● **Interrupt handling time**

After receiving an interrupt, the CPU requires 9 machine clock cycles to perform the following interrupt processing setup:

- Saves the program counter (PC) and program status (PS) values.
- Sets the PC to the start address (interrupt vector) of interrupt service routine.
- Updates the interrupt level bits (PS:CCR:IL1, IL0) in the program status (PS) register.

**Figure 8.1-4 Interrupt Processing Time**



When an interrupt request is generated immediately after the beginning of execution of the DIVU instruction requiring the longest execution cycle (17 machine clock cycles), it takes an interrupt processing time of  $17+9=26$  machine clock cycles.

The machine clock changes depending on the clock mode and main clock speed switching (gear function). For details, see "CHAPTER 6 CLOCK CONTROLLER".

## 8.1.5 Stack Operations During Interrupt Processing

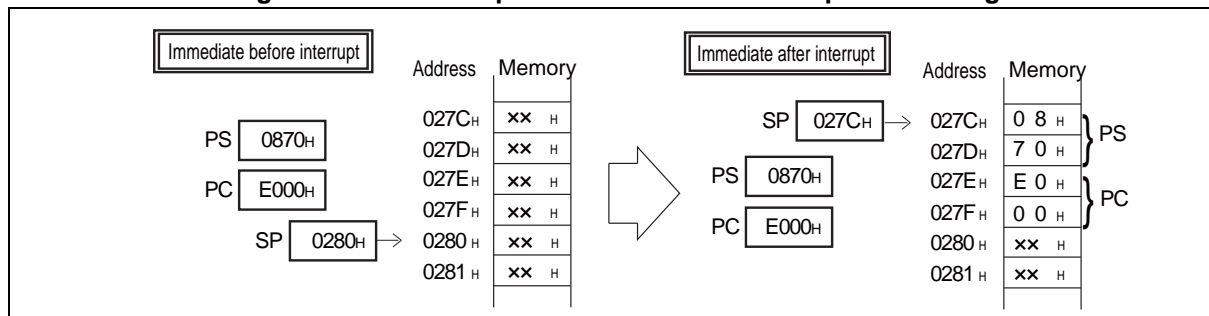
This section describes how registers are saved and restored during interrupt processing.

### ■ Stack Operation at the Start of Interrupt Processing

Once the CPU accepts an interrupt, it automatically saves the current program counter (PC) and program status (PS) values onto a stack.

Figure 8.1-5 shows how the stack is used at the start of interrupt processing.

**Figure 8.1-5 Stack Operation at Start of Interrupt Processing**



### ■ Stack Operation upon Returning from Interrupt

When the interrupt return instruction (RETI) is executed to end interrupt processing, the program status (PS) and then the program counter (PC) are restored from the stack, in the reverse order from which they were saved to the stack when interrupt processing started. This restores the PS and PC values to their states prior to starting interrupt processing.

Note:

As the accumulator (A) and temporary accumulator (T) are not saved onto the stack automatically, use the PUSHW and POPW instructions to save and restore the A and T values.

## MB95110B/M Series

### 8.1.6 Interrupt Processing Stack Area

The stack area in RAM is used for interrupt processing. The stack pointer (SP) contains the start address of the stack area.

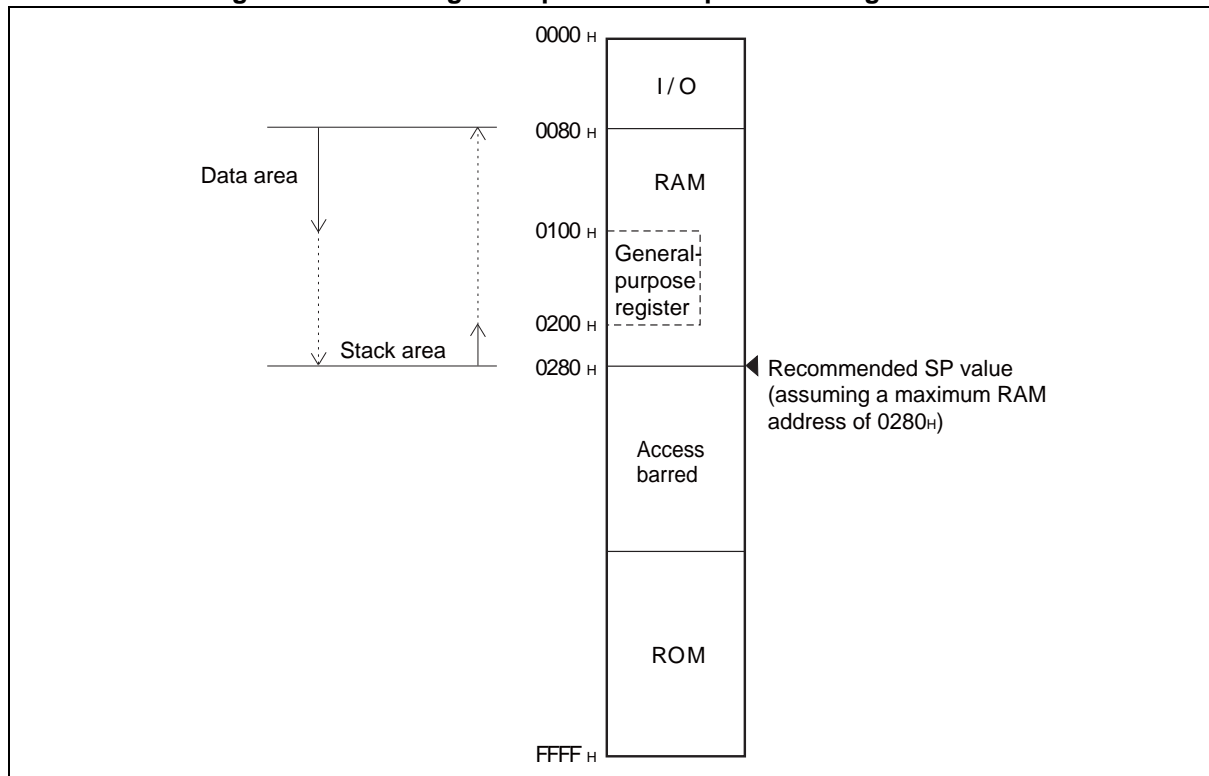
#### ■ Interrupt Processing Stack Area

The stack area is also used to save and restore the program counter (PC) when subroutine call (CALL) or vector call (CALLV) instructions are executed and to temporarily save and restore the registers via the PUSHW and POPW instructions.

- The stack area is located in RAM together with the data area.
- It is advisable to initialize the stack pointer (SP) to the maximum RAM address and allocate data areas starting from the minimum RAM address.

Figure 8.1-6 shows an example of setting the stack area.

**Figure 8.1-6 Setting Example of Interrupt Processing Stack Area**



#### Note:

The stack area is allocated in descending order of addresses for interrupts, subroutine calls, and the PUSHW instruction; it is deallocated in ascending order of addresses for return (PETI, RET) and POPW instructions. When the stack area address used decreases for nested interrupts or subroutines, prevent the stack area from overlapping the data area or general-purpose register area containing other data.



# **CHAPTER 9**

---

## ***I/O PORT***

**This chapter describes the functions and operations of the I/O ports.**

9.1 Overview of I/O Ports

9.2 Port 0

9.3 Port 1

9.4 Port 2

9.5 Port 3

9.6 Port 5

9.7 Port 6

9.8 Port G

## 9.1 Overview of I/O Ports

I/O ports are used to control general-purpose I/O pins.

### ■ Overview of I/O Ports

The I/O port has functions to output data from the CPU and load inputted signals into the CPU, via the port data register (PDR). It is also possible to set the input/output direction of the I/O pins as desired at the bit level, via the port direction register (DDR).

Table 9.1-1 lists the registers for each port.

**Table 9.1-1 Each Port Registers**

Register name		Read/Write	Initial value
Port 0 data register	(PDR0)	R, RM/W	00000000 <sub>B</sub>
Port 0 direction register	(DDR0)	R/W	00000000 <sub>B</sub>
Port 1 data register	(PDR1)	R, RM/W	00000000 <sub>B</sub>
Port 1 direction register	(DDR1)	R/W	00000000 <sub>B</sub>
Port 2 data register	(PDR2)	R, RM/W	00000000 <sub>B</sub>
Port 2 direction register	(DDR2)	R/W	00000000 <sub>B</sub>
Port 3 data register	(PDR3)	R, RM/W	00000000 <sub>B</sub>
Port 3 direction register	(DDR3)	R/W	00000000 <sub>B</sub>
Port 5 data register	(PDR5)	R, RM/W	00000000 <sub>B</sub>
Port 5 direction register	(DDR5)	R/W	00000000 <sub>B</sub>
Port 6 data register	(PDR6)	R, RM/W	00000000 <sub>B</sub>
Port 6 direction register	(DDR6)	R/W	00000000 <sub>B</sub>
Port G data register	(PDRG)	R, RM/W	00000000 <sub>B</sub>
Port G direction register	(DDRG)	R/W	00000000 <sub>B</sub>
Port 1 pull-up register	(PUL1)	R/W	00000000 <sub>B</sub>
Port 2 pull-up register	(PUL2)	R/W	00000000 <sub>B</sub>
Port 3 pull-up register	(PUL3)	R/W	00000000 <sub>B</sub>
Port G pull-up register	(PULG)	R/W	00000000 <sub>B</sub>
A/D input disable register lower	(AIDRL)	R/W	00000000 <sub>B</sub>
Input level selection register	(ILSR)	R/W	00000000 <sub>B</sub>
Input level selection register 2*	(ILSR2)	R/W	00000000 <sub>B</sub>

R/W: Readable/writable (Read value is the same as the write value.)

R, RM/W: Readable/writable (Read value is different from write value, write value is read by read-modify-write (RMW) instruction.)

\*: Only for 5V products, it is an effective register.

## 9.2 Port 0

**Port 0 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 0 Configuration

Port 0 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 0 data register (PDR0)
- Port 0 direction register (DDR0)
- Input level selection register 2 (ILSR2)

### ■ Port 0 Pins

Port 0 has eight I/O pins.

Table 9.2-1 lists the port 0 pins.

**Table 9.2-1 Port 0 Pins**

Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P00/INT00	P00 general-purpose I/O	INT00 external interrupt input	Hysteresis/automotive	CMOS	-	-
P01/INT01	P01 general-purpose I/O	INT01 external interrupt input	Hysteresis/automotive	CMOS	-	-
P02/INT02	P02 general-purpose I/O	INT02 external interrupt input	Hysteresis/automotive	CMOS	-	-
P03/INT03	P03 general-purpose I/O	INT03 external interrupt input	Hysteresis/automotive	CMOS	-	-
P04/INT04	P04 general-purpose I/O	INT04 external interrupt input	Hysteresis/automotive	CMOS	-	-
P05/INT05	P05 general-purpose I/O	INT05 external interrupt input	Hysteresis/automotive	CMOS	-	-
P06/INT06	P06 general-purpose I/O	INT06 external interrupt input	Hysteresis/automotive	CMOS	-	-
P07/INT07	P07 general-purpose I/O	INT07 external interrupt input	Hysteresis/automotive	CMOS	-	-

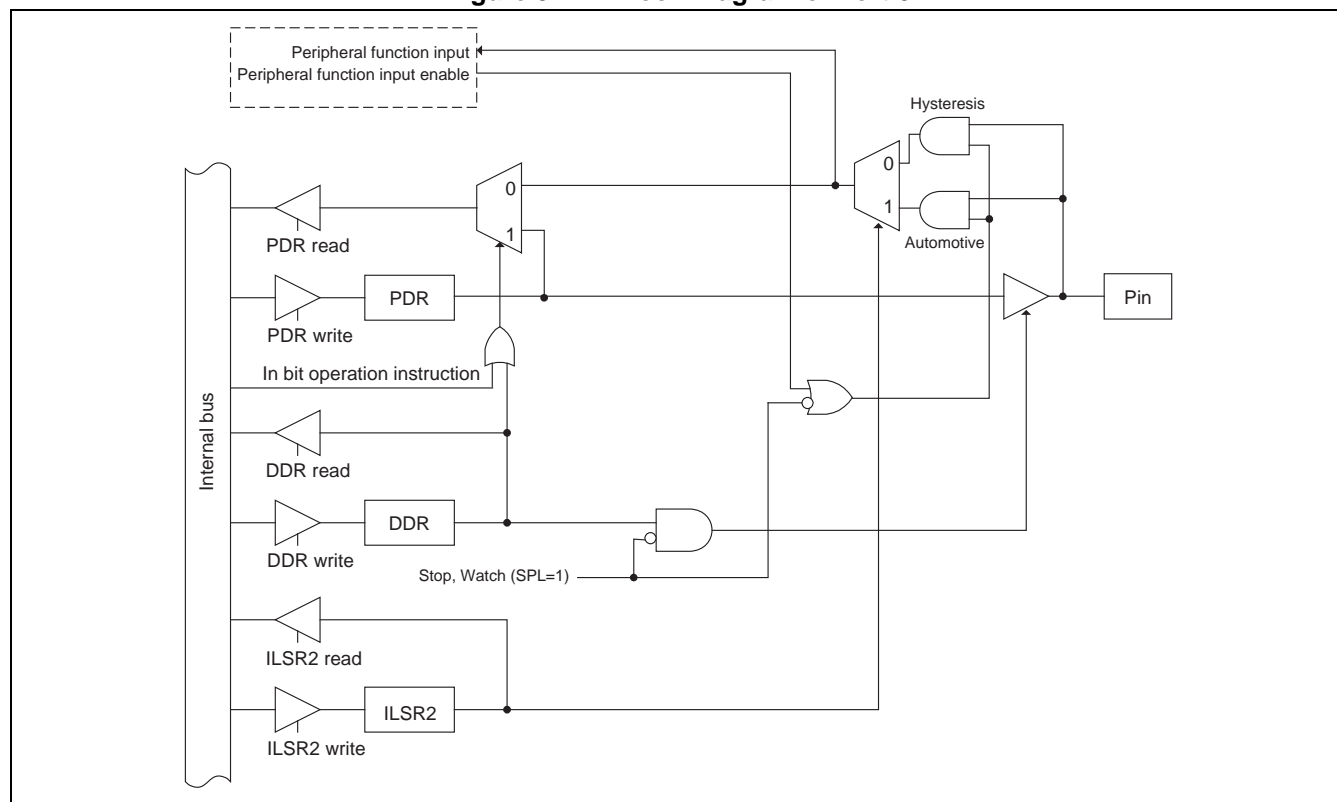
OD: Open drain, PU: Pull-up

\*:For 5V products, the hysteresis input can be switched to an automotive input. It becomes a hysteresis input besides.



# ■ Block Diagram of Port 0

Figure 9.2-1 Block Diagram of Port 0



## 9.2.1 Port 0 Registers

This section describes the port 0 registers.

### ■ Port 0 Register Function

Table 9.2-2 lists the port 0 register functions.

**Table 9.2-2 Port 0 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR0	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR0	0	Port input enabled		
	1	Port output enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.2-3 lists the correspondence between port 0 pins and each register bit.

**Table 9.2-3 Correspondence Between Registers and Pins for Port 0**

	Correspondence between related register bits and pins							
Pin name	P07	P06	P05	P04	P03	P02	P01	P00
PDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR0								
ILSR2*	bit0							

\*:Only for 5V products, it is an effective register.

## **9.2.2 Operations of Port 0**

---

**This section describes the operations of port 0.**

---

### **■ Operations of Port 0**

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

#### ● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

#### ● Operation of the external interrupt input pin

- Set the DDR register bit, which is corresponding to the external interrupt input pin, to "0".
- Pin values are continuously input to the external interrupt circuit. When using the pin for a function other than an interrupt, you must disable the corresponding external interrupt.

## ● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit0 of the ILSR2 register to "1" changes the port 0 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit0 of the ILSR2 register is "0".
- Only modify the port 0 input level setting when the peripheral function inputs are halted.

Table 9.2-4 shows the pin states of the port.

**Table 9.2-4 Pin State of Port 0**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff (If external interrupts are enabled, the external interrupt can be input.)	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

## 9.3 Port 1

**Port 1 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 1 Configuration

Port 1 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 1 data register (PDR1)
- Port 1 direction register (DDR1)
- Port 1 pull-up control register (PUL1)
- Input level selection register (ILSR)
- Input level selection register 2 (ILSR2)

### ■ Port 1 Pins

Port 1 has six I/O pins.

Table 9.3-1 lists the port 1 pins.

**Table 9.3-1 Port 1 Pins**

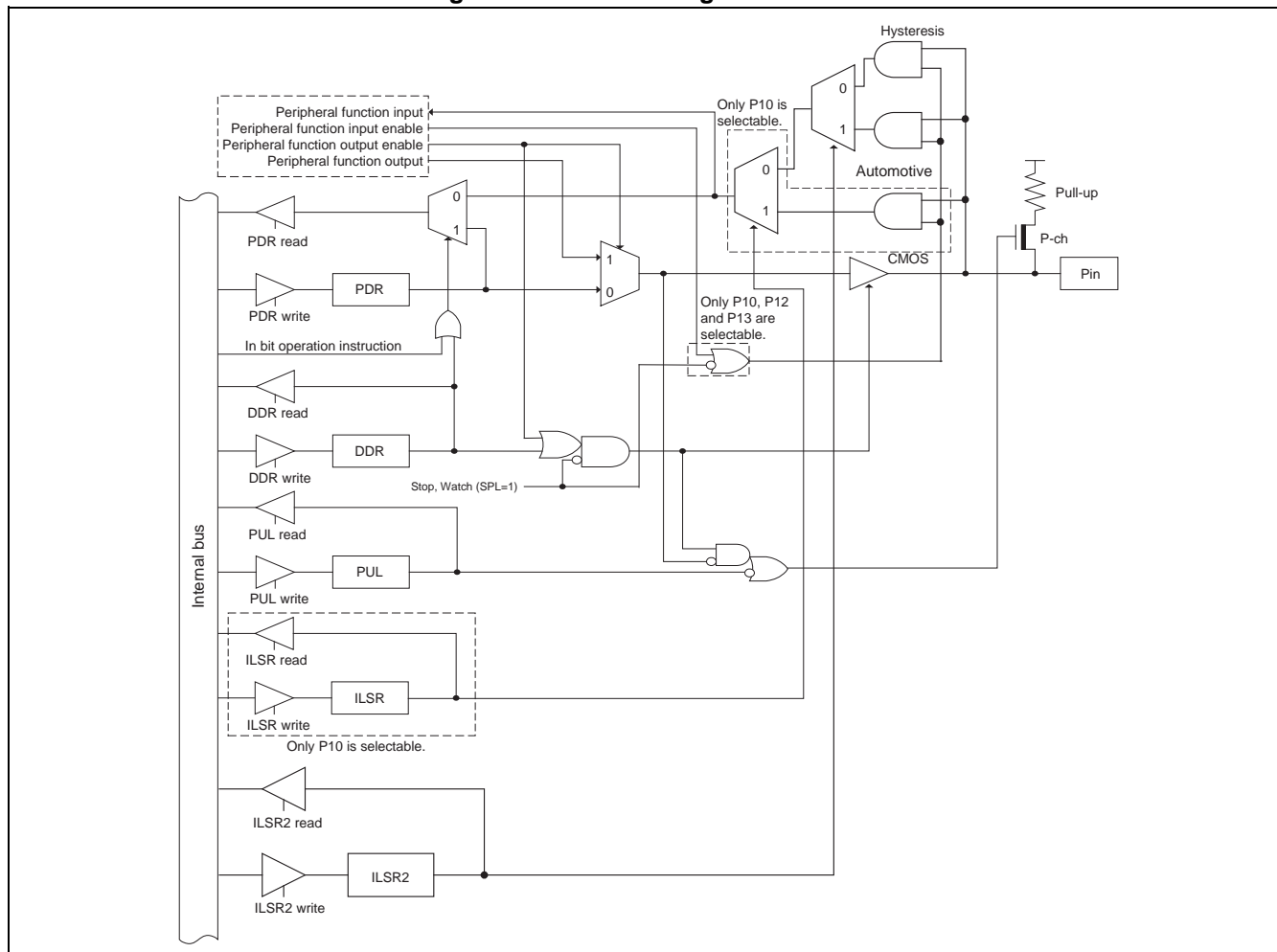
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P10/UI0	P10 general-purpose I/O	UI0 UART/SIO ch.0 data input	Hysteresis/CMOS/ automotive	CMOS	-	○
P11/UO0	P11 general-purpose I/O	UO0 UART/SIO ch.0 data output	Hysteresis/automotive	CMOS	-	○
P12/UCK0	P12 general-purpose I/O	UO0 UART/SIO ch.0 clock I/O	Hysteresis/automotive	CMOS	-	○
P13/ TRG0/ADTG	P13 general-purpose I/O	TRG0 16-bit PPG ch.0 trigger input	Hysteresis/automotive	CMOS	-	○
		ADTG A/D trigger activation input				
P14/PPG0	P14 general-purpose I/O	PPG0 16-bit PPG ch.0 output	Hysteresis/automotive	CMOS	-	○
P15	P15 general-purpose I/O	Not shared	Hysteresis/automotive	CMOS	-	○

OD: Open drain, PU: Pull-up

\*: Only for 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input or CMOS input besides.

## ■ Block Diagram of Port 1

Figure 9.3-1 Block Diagram of Port 1



## 9.3.1 Port 1 Registers

This section describes the port 1 registers.

### ■ Port 1 Register Function

Table 9.3-2 lists the port 1 register functions.

**Table 9.3-2 Port 1 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR1	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR1	0	Port input enabled		
	1	Port output enabled		
PUL1	0	Pull-up disabled		
	1	Pull-up enabled		
ILSR	0	Hysteresis input level selection		
	1	CMOS input level selection		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.3-3 lists the correspondence between port 1 pins and each register bit.

**Table 9.3-3 Correspondence Between Registers and Pins for Port 1**

	Correspondence between related register bits and pins							
Pin name	-	-	P15	P14	P13	P12	P11	P10
PDR1	-	-	bit5	bit4	bit3	bit2	bit1	bit0
DDR1								
PUL1								
ILSR	-	-	-	-	-	-	-	bit0
ILSR2*	-	-	bit1					

\*: Only for 5V products, it is an effective register.

## 9.3.2 Operations of Port 1

---

**This section describes the operations of port 1.**

---

### ■ Operations of Port 1

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.



● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input of P10/UI0, P12/UCK0 and P12/TRG0/ADTG port is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", set the DDR register bit, the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

● Operation of the input level selection register

- Writing "1" to the bit0 of ILSR register changes only P10 from the hysteresis input level to the CMOS input level. When the bit0 of ILSR register is "0", it should be the hysteresis input level.
- For pins other than P10, the CMOS input level cannot be selected; however, only the hysteresis input level or the automotive input level can.
- Make sure that the input level for P10 is changed during the peripheral function (UART/SIO) stopped.

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models. Setting bit1 of the ILSR2 register to "1" changes the port 1 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit1 of the ILSR2 register is "0".
- P10 only uses the automotive input level when bit0 of the ILSR register is "0". In the case of P10 only, setting "1" to bit0 of the ILSR register has priority over ILSR2.
- Only modify the port 1 input level setting when the peripheral functions (ART/SIO) are halted.

Table 9.3-4 shows the pin states of the port.

**Table 9.3-4 Pin State of Port 1**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z (the pull-up setting is enabled) Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

## 9.4 Port 2

**Port 2 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 2 Configuration

Port 2 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 2 data register (PDR2)
- Port 2 direction register (DDR2)
- Port 2 pull-up control register (PUL2)
- Input level selection register 2 (ILSR2)

### ■ Port 2 Pins

Port 2 has five I/O pins.

Table 9.4-1 lists the port 2 pins.

**Table 9.4-1 Port 2 Pins**

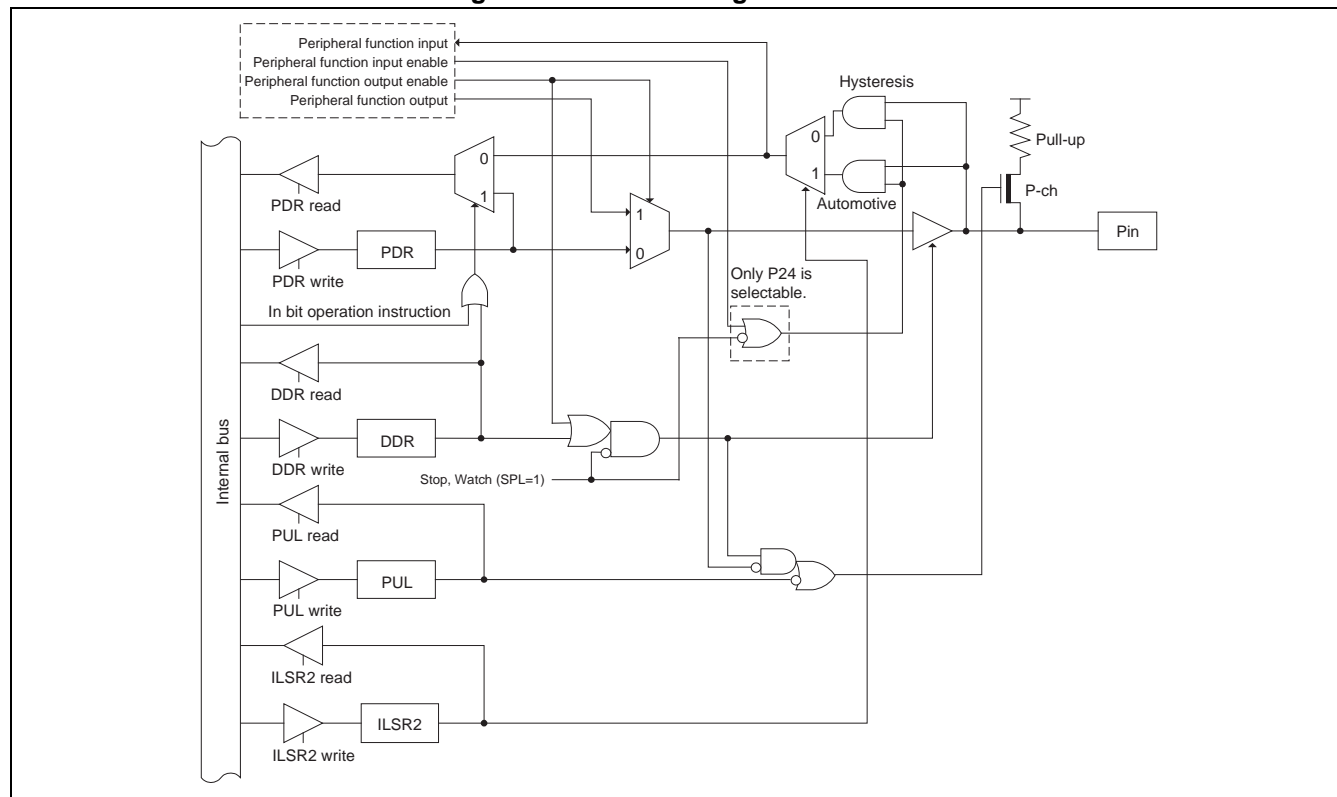
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P20/PPG00	P20 general-purpose I/O	PPG00 8/16-bit PPG0 ch.0 data output	Hysteresis/automotive	CMOS	-	○
P21/PPG01	P21 general-purpose I/O	PPG01 8/16-bit PPG0 ch.1 data output	Hysteresis/automotive	CMOS	-	○
P22/TO00	P22 general-purpose I/O	TO00 8/16-bit compound timer 00 output	Hysteresis/automotive	CMOS	-	○
P23/TO01	P23 general-purpose I/O	TO01 8/16-bit compound timer 01 output	Hysteresis/automotive	CMOS	-	○
P24/EC0	P24 general-purpose I/O	EC0 8/16-bit compound timer ch.0 external clock input	Hysteresis/automotive	CMOS	-	○

OD: Open drain, PU: Pull-up

\*:For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

## ■ Block Diagram of Port 2

Figure 9.4-1 Block Diagram of Port 2



## 9.4.1 Port 2 Registers

This section describes the port 2 registers.

### ■ Port 2 Register Function

Table 9.4-2 lists the port 2 register functions.

**Table 9.4-2 Port 2 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR2	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR2	0	Port input enabled		
	1	Port output enabled		
PUL2	0	Pull-up disabled		
	1	Pull-up enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.4-3 lists the correspondence between port 2 pins and each register bit.

**Table 9.4-3 Correspondence Between Registers and Pins for Port 2**

	Correspondence between related register bits and pins							
Pin name	-	-	-	P24	P23	P22	P21	P20
PDR2	-	-	-	bit4	bit3	bit2	bit1	bit0
DDR2								
PUL2								
ILSR2*	-	-	-	bit2				

\*: Only for 5V products, it is an effective register.

## **9.4.2 Operations of Port 2**

---

**This section describes the operations of port 2.**

---

### **■ Operations of Port 2**

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

## ● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input of P24/EC0 port is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

## ● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

## ● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V products.
- Setting bit2 of the ILSR2 register to "1" changes the port 2 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit2 of the ILSR2 register is "0".
- Only modify the port 2 input level setting when the peripheral function inputs are halted.

Table 9.4-4 shows the pin states of the port.

**Table 9.4-4 Pin State of Port 2**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z (the pull-up setting is enabled) Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

## 9.5 Port 3

**Port 3 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 3 Configuration

Port 3 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 3 data register (PDR3)
- Port 3 direction register (DDR3)
- Port 3 pull-up control register (PUL3)
- A/D input disable register low (AIDRL)
- Input level selection register 2 (ILSR2)

### ■ Port 3 Pins

Port 3 has eight I/O pins.

Table 9.5-1 lists the port 3 pins.

**Table 9.5-1 Port 3 Pins**

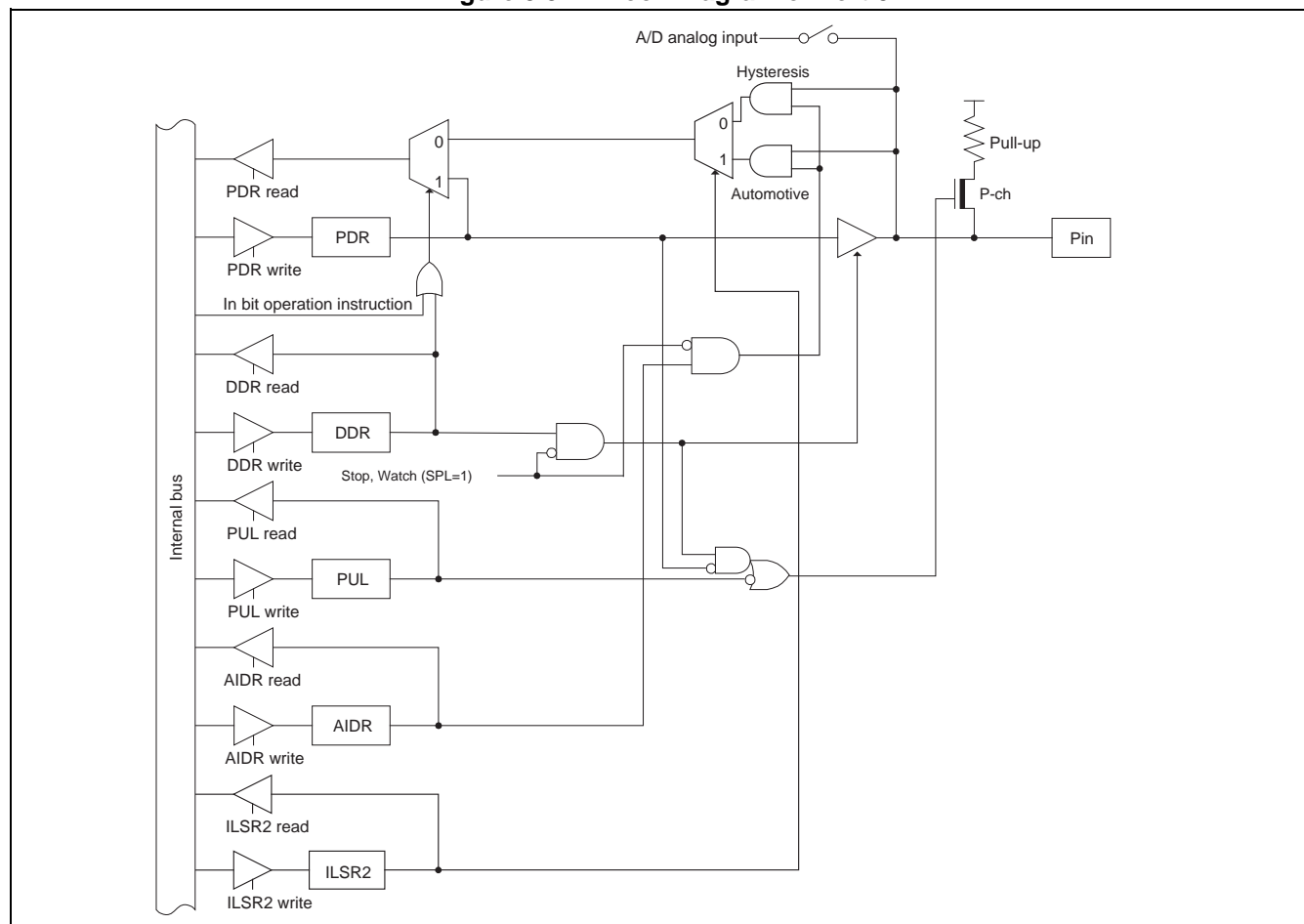
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P30/AN00	P30 general-purpose I/O	AN00 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P30/AN00	P31 general-purpose I/O	AN01 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P32/AN02	P32 general-purpose I/O	AN02 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P33/AN03	P33 general-purpose I/O	AN03 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P34/AN04	P34 general-purpose I/O	AN04 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P35/AN05	P35 general-purpose I/O	AN05 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P36/AN06	P36 general-purpose I/O	AN06 analog input	Hysteresis/automotive/ analog	CMOS	-	○
P37/AN07	P37 general-purpose I/O	AN07 analog input	Hysteresis/automotive/ analog	CMOS	-	○

OD: Open drain, PU: Pull-up

\*:For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

## ■ Block Diagram of Port 3

Figure 9.5-1 Block Diagram of Port 3





## 9.5.1 Port 3 Registers

This section describes the port 3 registers.

### ■ Port 3 Register Function

Table 9.5-2 lists the port 3 register functions.

**Table 9.5-2 Port 3 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR3	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR3	0	Port input enabled		
	1	Port output enabled		
PUL3	0	Pull-up disabled		
	1	Pull-up enabled		
AIDRL	0	Analog input enabled		
	1	Port input enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.5-3 lists the correspondence between port 3 pins and each register bit.

**Table 9.5-3 Correspondence Between Registers and Pins for Port 3**

	Correspondence between related register bits and pins							
Pin name	P37	P36	P35	P34	P33	P32	P31	P30
PDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR3								
PUL3								
AIDRL								
ILSR2*								

\*: Only for 5V products, it is an effective register.

## 9.5.2 Operations of Port 3

---

**This section describes the operations of port 3.**

---

### ■ Operations of Port 3

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- When using the analog input shared pin as an input port, set the corresponding bits in the A/D input disable register low (AIDRL) to "1".
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register and AIDRL register values to "0", and sets the port input disabled.

#### ● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

#### ● Operation as an analog input

- Set the DDR register bit, which is corresponding to the analog input pin, to "0", and set the AIDRL register bit to "0".
- Set the corresponding PUL register bit to "0".

● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit3 of the ILSR2 register to "1" changes the port 3 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit3 of the ILSR2 register is "0".

Table 9.5-4 shows the pin states of the port.

**Table 9.5-4 Pin State of Port 3**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ analog input	Hi-Z (the pull-up setting is enabled) Input cutoff	Hi-Z Input disabled*

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input disabled" means the state that the operation of the input gate close to the pin is disabled.

## 9.6 Port 5

**Port 5 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 5 Configuration

Port 5 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 5 data register (PDR5)
- Port 5 direction register (DDR5)
- Input level selection register (ILSR)
- Input level selection register 2 (ILSR2)

### ■ Port 5 Pins

Port 5 has two I/O pins.

Table 9.6-1 lists the port 5 pins.

**Table 9.6-1 Port 5 Pins**

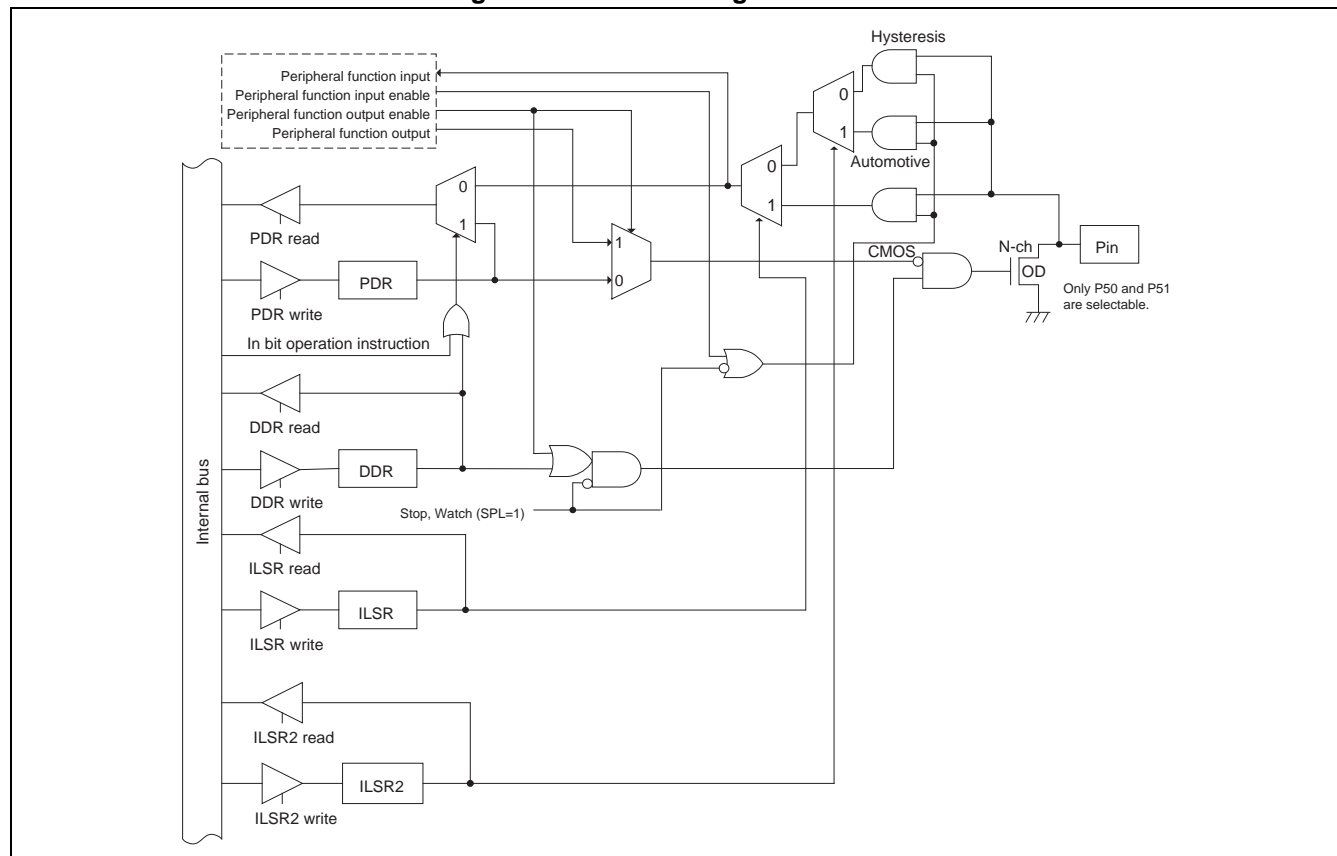
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P50/SCL0	P50 general-purpose I/O	SCL0 I <sup>2</sup> C ch.0 clock I/O	Hysteresis/CMOS/ automotive	CMOS	○	-
P51/SDA0	P51 general-purpose I/O	SDA0 I <sup>2</sup> C ch.0 data I/O	Hysteresis/CMOS/ automotive	CMOS	○	-

OD: Open drain, PU: Pull-up

\*:For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input or CMOS input besides.

# ■ Block Diagram of Port 5

Figure 9.6-1 Block Diagram of Port 5



## 9.6.1 Port 5 Registers

This section describes the port 5 registers.

### ■ Port 5 Register Function

Table 9.6-2 lists the port 5 register functions.

**Table 9.6-2 Port 5 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR5	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level <sup>*1</sup> .
DDR5	0	Port input enabled		
	1	Port output enabled		
ILSR	0	Hysteresis input level selection		
	1	CMOS input level selection		
ILSR2 <sup>*2</sup>	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*1: For N-ch open drain pin, this should be Hi-Z.

\*2: Only for 5V products, it is an effective register.

Table 9.6-3 lists the correspondence between port 5 pins and each register bit.

**Table 9.6-3 Correspondence Between Registers and Pins for Port 5**

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	-	-	P51	P50
PDR5	-	-	-	-	-	-	bit1	bit0
DDR5	-	-	-	-	-	-	bit4	bit3
ILSR	-	-	-	-	-	-	bit4	bit3
ILSR2 <sup>*</sup>	-	-	-	-	-	-	bit4	

\*: Only for 5V products, it is an effective register.

## 9.6.2 Operations of Port 5

---

**This section describes the operations of port 5.**

---

### ■ Operations of Port 5

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

## ● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the peripheral function input (SCL0, SDA0) is enabled, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

## ● Operation of the input level selection register

- Setting "1" to the bit4 and bit3 of ILSR register changes only P51 and P50 from the hysteresis input level to the CMOS input level. When the bit4 and bit3 of ILSR register is "0", it should be the hysteresis input level.
- Make sure that the input level for P51 and P50 is changed during the peripheral function (I<sup>2</sup>C) stopped.

## ● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit4 of the ILSR2 register to "1" changes the port 5 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit4 of the ILSR2 register is "0".
- Only modify the port 5 input level setting when the peripheral function (I<sup>2</sup>C) is halted.
- P51 and P50 only use the automotive input level when bits 4 and 3 of the ILSR register are "0". Setting "1" to bits 4 and 3 of the ILSR register has priority over the ILSR2 register.

Table 9.6-4 shows the pin states of the port.

**Table 9.6-4 Pin State of Port 5**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z (the pull-up setting is enabled) Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.



## 9.7 Port 6

**Port 6 is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

**See the chapters on each peripheral function for details about peripheral functions.**

### ■ Port 6 Configuration

Port 6 is made up of the following elements.

- General-purpose I/O pins/peripheral function I/O pins
- Port 6 data register (PDR6)
- Port 6 direction register (DDR6)
- Input level selection register (ILSR)
- Input level selection register 2 (ILSR2)

### ■ Port 6 Pins

Port 6 has eight I/O pins.

Table 9.7-1 lists the port 6 pins.

**Table 9.7-1 Port 6 Pins**

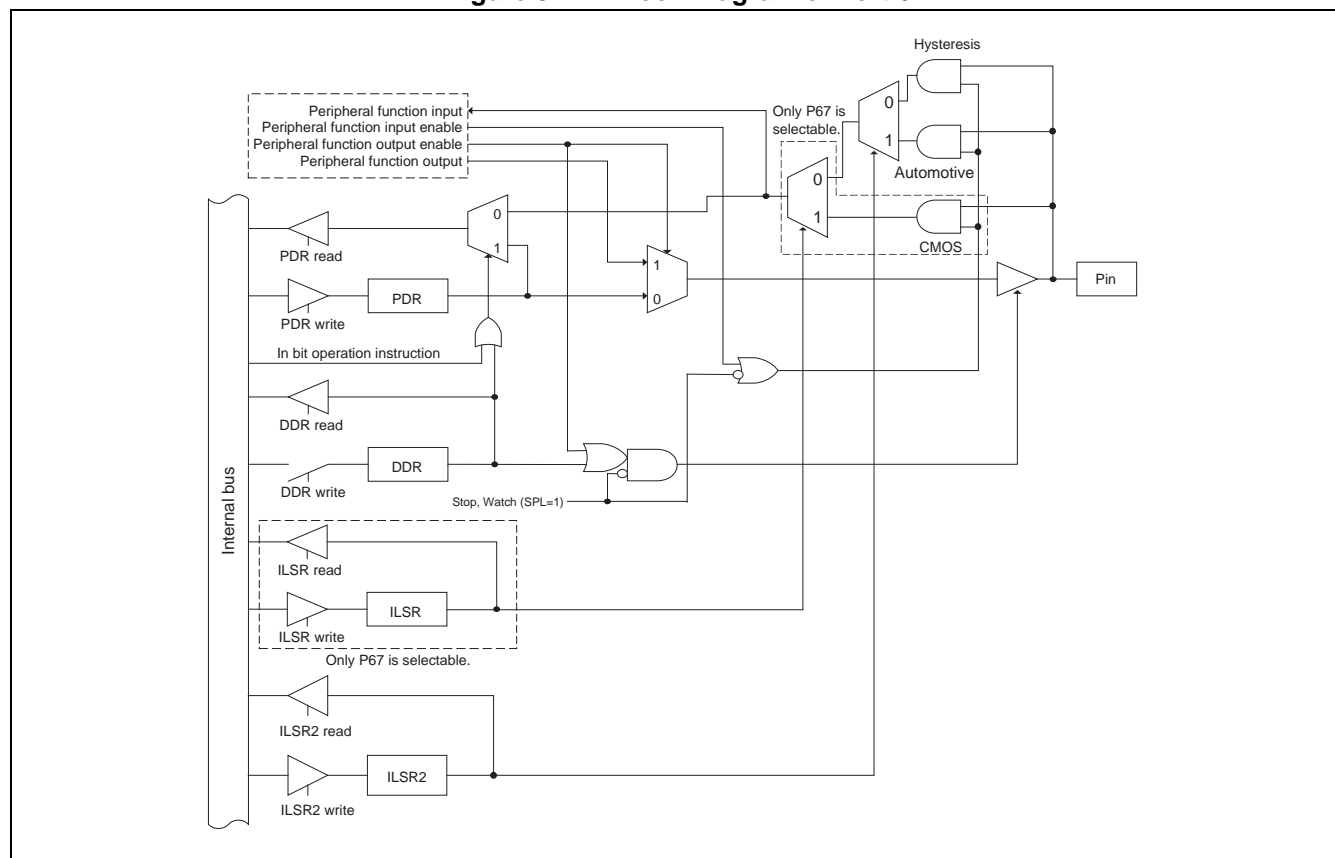
Pin name	Function	Shared peripheral functions	I/O type			
			Input*	Output	OD	PU
P60/PPG10	P60 general-purpose I/O	PPG10 8/16-bit PPG1 ch.0 output	Hysteresis/automotive	CMOS	-	-
P61/PPG11	P61 general-purpose I/O	PPG11 8/16-bit PPG1 ch.1 output	Hysteresis/automotive	CMOS	-	-
P62/TO10	P62 general-purpose I/O	TO10 8/16-bit compound timer 10 output	Hysteresis/automotive	CMOS	-	-
P63/TO11	P63 general-purpose I/O	TO11 8/16-bit compound timer 11 output	Hysteresis/automotive	CMOS	-	-
P64/EC1	P64 general-purpose I/O	EC1 8/16-bit compound timer ch.1 clock input	Hysteresis/automotive	CMOS	-	-
P65/SCK	P65 general-purpose I/O	LIN-UART clock I/O	Hysteresis/automotive	CMOS	-	-
P66/SOT	P66 general-purpose I/O	LIN-UART data output	Hysteresis/automotive	CMOS	-	-
P67/SIN	P67 general-purpose I/O	LIN-UART data input	Hysteresis/CMOS/automotive	CMOS	-	-

OD: Open drain, PU: Pull-up

\*:For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input or CMOS input besides.

## ■ Block Diagram of Port 6

Figure 9.7-1 Block Diagram of Port 6



## 9.7.1 Port 6 Registers

This section describes the port 6 registers.

### ■ Port 6 Register Function

Table 9.7-2 lists the port 6 register functions.

**Table 9.7-2 Port 6 Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDR6	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDR6	0	Port input enabled		
	1	Port output enabled		
ILSR	0	Hysteresis input level selection		
	1	CMOS input level selection		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.7-3 lists the correspondence between port 6 pins and each register bit.

**Table 9.7-3 Correspondence Between Registers and Pins for Port 6**

	Correspondence between related register bits and pins							
Pin name	P67	P66	P65	P64	P63	P62	P61	P60
PDR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DDR6								
ILSR	bit2	-	-	-	-	-	-	-
ILSR2*	bit5							

\*: Only for 5V products, it is an effective register.

## 9.7.2 Operations of Port 6

---

**This section describes the operations of port 6.**

---

### ■ Operations of Port 6

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- For a peripheral function sharing pins, disable its output.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- For a peripheral function sharing pins, disable its output.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation as a peripheral function output

- Setting the output enable bit of a peripheral function sets the corresponding pin as a peripheral function output.
- The pin value can be read from the PDR register even if the peripheral function output is enabled. Therefore, the output value of a peripheral function can be read by the read operation on PDR register. However, the read-modify-write (RMW) command returns the PDR register value.

#### ● Operation as a peripheral function input

- Set the DDR register bit, which is corresponding to the peripheral function input pin, to "0" to set a pin as an input port.
- Reading the PDR register returns the pin value, regardless of whether the peripheral function uses an input pin. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input. However, if the interrupt input of P65/SCK and P67/SIN port is enabled for the external interrupt control register (EIC) of the external interrupt circuit and the interrupt pin selection circuit control register (WICR) of the external interrupt selection circuit, the input is enabled and not blocked.
- If the pin state specification bit is "0", the state remains in port I/O or peripheral function I/O and the output is maintained.

● Operation of the input level selection register

- Setting "1" to the bit2 of ILSR register changes only P67 from the hysteresis input level to the CMOS input level. When the bit2 of ILSR register is "0", it should be the hysteresis input level.
- For pins other than P67, the CMOS input level cannot be selected. Only the hysteresis input level or automotive input level can be selected.
- Make sure that the input level for P67 is changed during the peripheral function (LIN-UART) stopped.

● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit5 of the ILSR2 register to "1" changes the port 6 input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit5 of the ILSR2 register is "0".
- Only modify the port 6 input level setting when the peripheral function (LIN-UART) is halted.
- P67 only uses the automotive input level when bit2 of the ILSR register is "0". Setting "1" to bit2 of the ILSR register has priority over the ILSR2 register.

Table 9.7-4 shows the pin states of the port.

**Table 9.7-4 Pin State of Port 6**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port/ peripheral function I/O	Hi-Z Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.

## MB95110B/M Series

### 9.8 Port G

**Port G is a general-purpose I/O port.**

**This section focuses on functions as a general-purpose I/O port.**

#### ■ Port G Configuration

Port G is made up of the following elements.

- General-purpose I/O pins
- Port G data register (PDRG)
- Port G direction register (DDRG)
- Port G pull-up control register (PULG)
- Input level selection register 2 (ILSR2)

#### ■ Port G Pins

Port G has three I/O pins.

Table 9.8-1 lists the port G pins.

**Table 9.8-1 Port G Pins**

Pin name	Function	Shared peripheral functions	I/O type			
			Input <sup>*3</sup>	Output	OD	PU
PG0 <sup>*1</sup>	PG0 general-purpose I/O	Not shared	Hysteresis/automotive	CMOS	-	○
PG1 <sup>*2</sup>	PG1 general-purpose I/O	Not shared	Hysteresis/automotive	CMOS	-	○
PG2 <sup>*2</sup>	PG2 general-purpose I/O	Not shared	Hysteresis/automotive	CMOS	-	○

OD: Open drain, PU: Pull-up

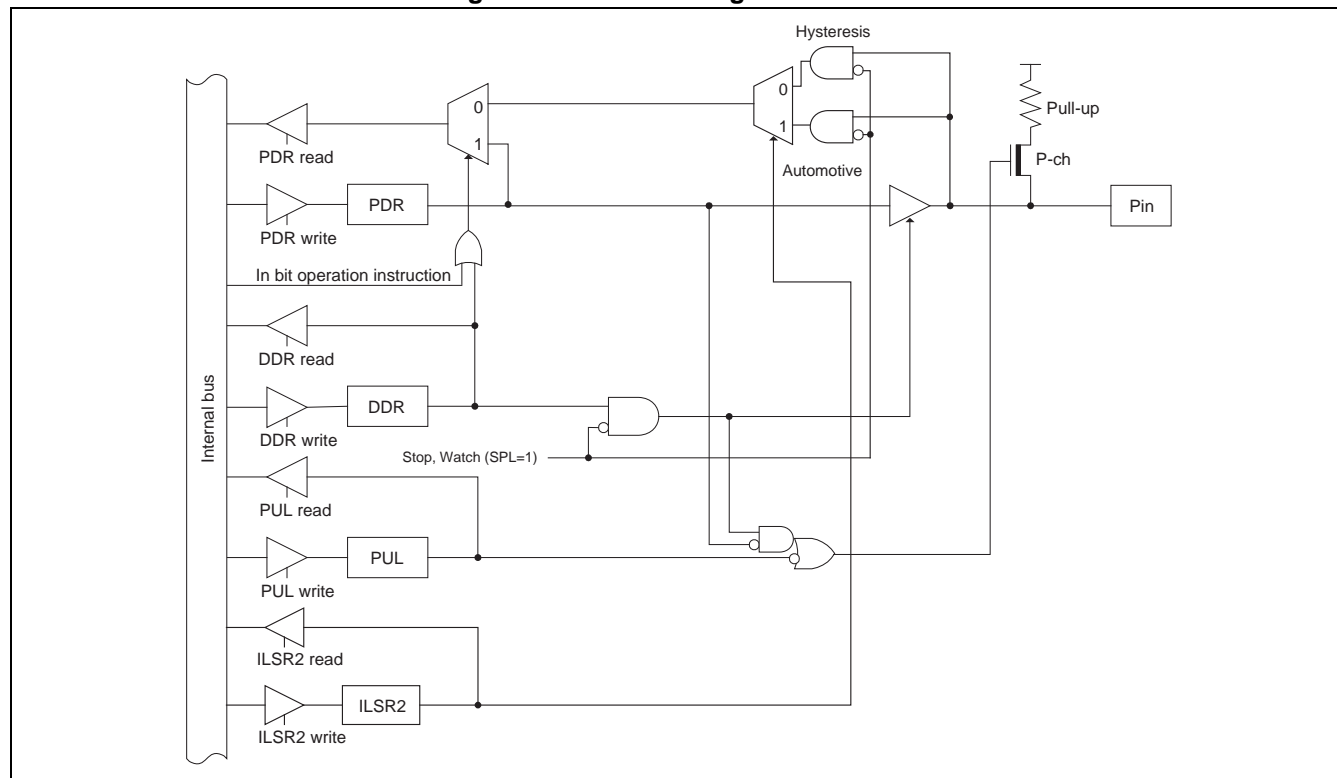
\*1:For the 5V product, the C pin is used.

\*2:For the single clock product, the general-purpose port is used; for the dual clock product, the sub clock oscillation pin is used.

\*3:For 5V products, the hysteresis input can be switched to the automotive input. It becomes hysteresis input besides.

## ■ Block Diagram of Port G

Figure 9.8-1 Block Diagram of Port G



## 9.8.1 Port G Registers

This section describes the port G registers.

### ■ Port G Register Function

Table 9.8-2 lists the port G register functions.

**Table 9.8-2 Port G Register Function**

Register name	Data	Read	Read read-modify-write	Write
PDRG	0	Pin state is "L" level.	PDR register value is "0".	As output port, outputs "L" level.
	1	Pin state is "H" level.	PDR register value is "1".	As output port, outputs "H" level.
DDRG	0	Port input enabled		
	1	Port output enabled		
PULG	0	Pull-up disabled		
	1	Pull-up enabled		
ILSR2*	0	Hysteresis input level selection		
	1	Automotive input level selection		

\*: Only for 5V products, it is an effective register.

Table 9.8-3 lists the correspondence between port G pins and each register bit.

**Table 9.8-3 Correspondence Between Registers and Pins for Port G**

	Correspondence between related register bits and pins							
Pin name	-	-	-	-	-	PG2	PG1	PG0 <sup>*2</sup>
PDRG	-	-	-	-	-	bit2	bit1	bit0
DDRG								
PULG								
ILSR2 <sup>*1</sup>	-	-	-	-	-	bit6		

\*1: Only for 5V products, it is an effective register.

\*2: For the 5V product, the C pin is used.



## 9.8.2 Operations of Port G

---

**This section describes the operations of port G.**

---

### ■ Operations of Port G

#### ● Operation as an output port

- Setting the corresponding DDR register bit to "1" sets a pin as an output port.
- When a pin is set as an output port, it outputs the value of the PDR register to pins.
- If data is written to the PDR register, the value is stored in the output latch and output to the pin as it is.
- Reading the PDR register returns the PDR register value.

#### ● Operation as an input port

- Setting the corresponding DDR register bit to "0" sets a pin as an input port.
- If data is written to the PDR register, the value is stored in the output latch but not output to the pin.
- Reading the PDR register returns the pin value. However, the read-modify-write (RMW) instruction returns the PDR register value.

#### ● Operation at reset

Resetting the CPU initializes the DDR register values to "0", and sets the port input enabled.

#### ● Operation in stop mode and watch mode

- If the pin state specification bit in the standby control register (STBC:SPL) is set to "1" when the device switches to stop or watch mode, the pin is set forcibly to the high-impedance state regardless of the DDR register value.  
Note that the input is locked to "L" level and blocked in order to prevent leaks due to freed input.
- If the pin state specification bit is "0", the state remains in port I/O and the output is maintained.

#### ● Operation of the pull-up control register

Setting "1" to the PUL register connects the pull-up resistor to the pin.

However, when the general-purpose I/O port or shared peripheral resource outputs "L" level, the pull-up resistor is disconnected regardless of the PUL register value.

#### ● Operation of input level selection register 2

- The ILSR2 register is a valid register only for 5V models.
- Setting bit6 of the ILSR2 register to "1" changes the port G input level from the hysteresis input level to the automotive input level. The hysteresis input level is used when bit6 of the ILSR2 register is "0".

Table 9.8-4 shows the pin states of the port.

**Table 9.8-4 Pin State of Port G**

Operating state	Normal operation Sleep Stop (SPL=0) Watch (SPL=0)	Stop (SPL=1) Watch (SPL=1)	At reset
Pin state	I/O port	Hi-Z Input cutoff	Hi-Z Input enabled* (Not functional)

SPL: Pin state specification bit in standby control register (STBC:SPL)

Hi-Z: High impedance

\*: "Input enabled" means that the input function is in the enabled state. After reset, setting for internal pull-up or output pin is recommended.



# ***CHAPTER 10***

---

## ***TIME-BASE TIMER***

**This chapter describes the functions and operations of the time-base timer.**

- 10.1 Overview of Time-base Timer
- 10.2 Configuration of Time-base Timer
- 10.3 Registers of the Time-base Timer
- 10.4 Interrupts of Time-base Timer
- 10.5 Explanation of Time-base Timer Operations and Setup Procedure Example
- 10.6 Notes on Using Time-base Timer

## 10.1 Overview of Time-base Timer

The time-base timer is a 22-bit free-run down-counting counter which is synchronized with the main clock divided by two. The time-base timer has an interval timer function which can repeatedly generate interrupt requests at regular intervals.

### ■ Interval Timer Function

The interval timer function repeatedly generates interrupt requests at regular intervals by using the main clock divided by two as the count clock.

- The counter of the time-base timer counts down so that an interrupt request is generated every time the selected interval time elapses.
- The interval time can be selected from the following four types.

Table 10.1-1 shows the interval times available to the time-base timer.

**Table 10.1-1 Interval Times of Time-base Timer**

Internal count clock cycle	Interval time
$2/F_{CH}(0.5\ \mu s)$	$2^{10} \times 2/F_{CH}(512.0\ \mu s)$
	$2^{12} \times 2/F_{CH}(2.05ms)$
	$2^{14} \times 2/F_{CH}(8.19ms)$
	$2^{16} \times 2/F_{CH}(32.77ms)$

$F_{CH}$ : Main clock

The values in parentheses represent the values used when the main clock operates at 4MHz.

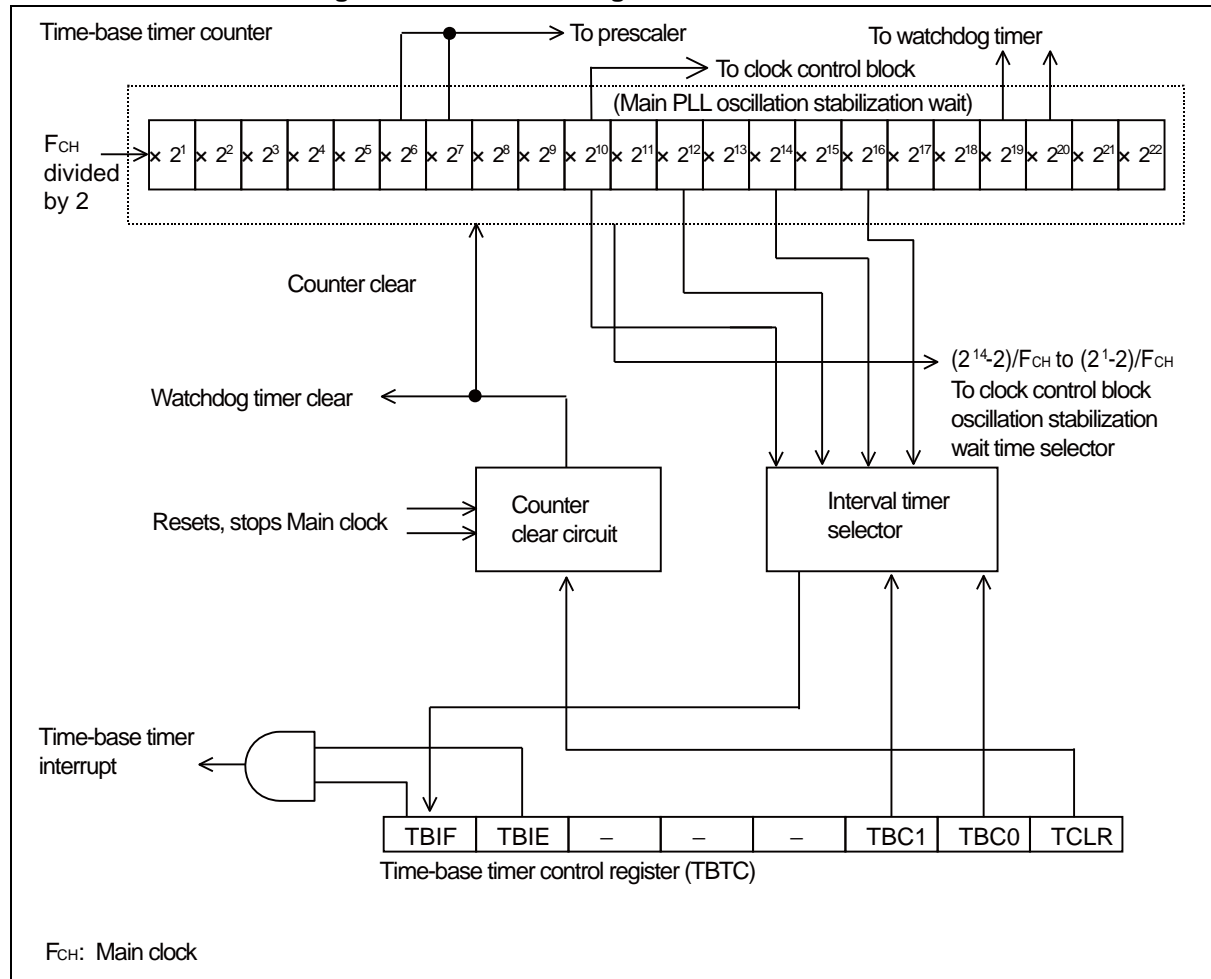
## 10.2 Configuration of Time-base Timer

The time-base timer consists of the following blocks:

- Time-base timer counter
- Counter clear circuit
- Interval timer selector
- Time-base Timer Control Register (TBTC)

### ■ Block Diagram of Time-base Timer

Figure 10.2-1 Block Diagram of Time-base Timer



- Time-base timer counter

22-bit down-counter that uses the main clock divided by two as the count clock.

- Counter clear circuit

This circuit controls clearing of the time-base counter.

- Interval timer selector

This circuit selects the one bit from four bits in the 22 bits that make up the time-base timer counter to use the interval timer.

- Time-base timer control register (TBTC)

This register selects the interval time, clears the counter, controls interrupts and checks the status.

■ **Input Clock**

The time-base timer uses the main clock divided by two as its input clock (count clock).

■ **Output Clock**

The time-base timer supplies clocks to the main clock oscillation stabilization wait time timer, the watchdog timer and the prescaler.

## 10.3 Registers of the Time-base Timer

Figure 10.3-1 shows the register of the Time-base Timer.

### ■ Registers of the Time-base Timer

**Figure 10.3-1 Register of the Time-base Timer**

Time-base timer control register (TBTC)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000A <sub>H</sub>	TBIF	TBIE	–	–	–	TBC1	TBC0	TCLR	00000000 <sub>B</sub>
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

R/W : Readable/writable (Read value is the same as write value)

R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

R0,W : Write only (Writable, "0" is read)

R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

– : Undefined



## 10.3.1 Time-base Timer Control Register (TBTC)

The time-base timer control register (TBTC) selects the interval time, clears the counter, controls interrupts and checks the status.

### ■ Time-base Timer Control Register (TBTC)

Figure 10.3-2 Time-base Timer Control Register (TBTC)

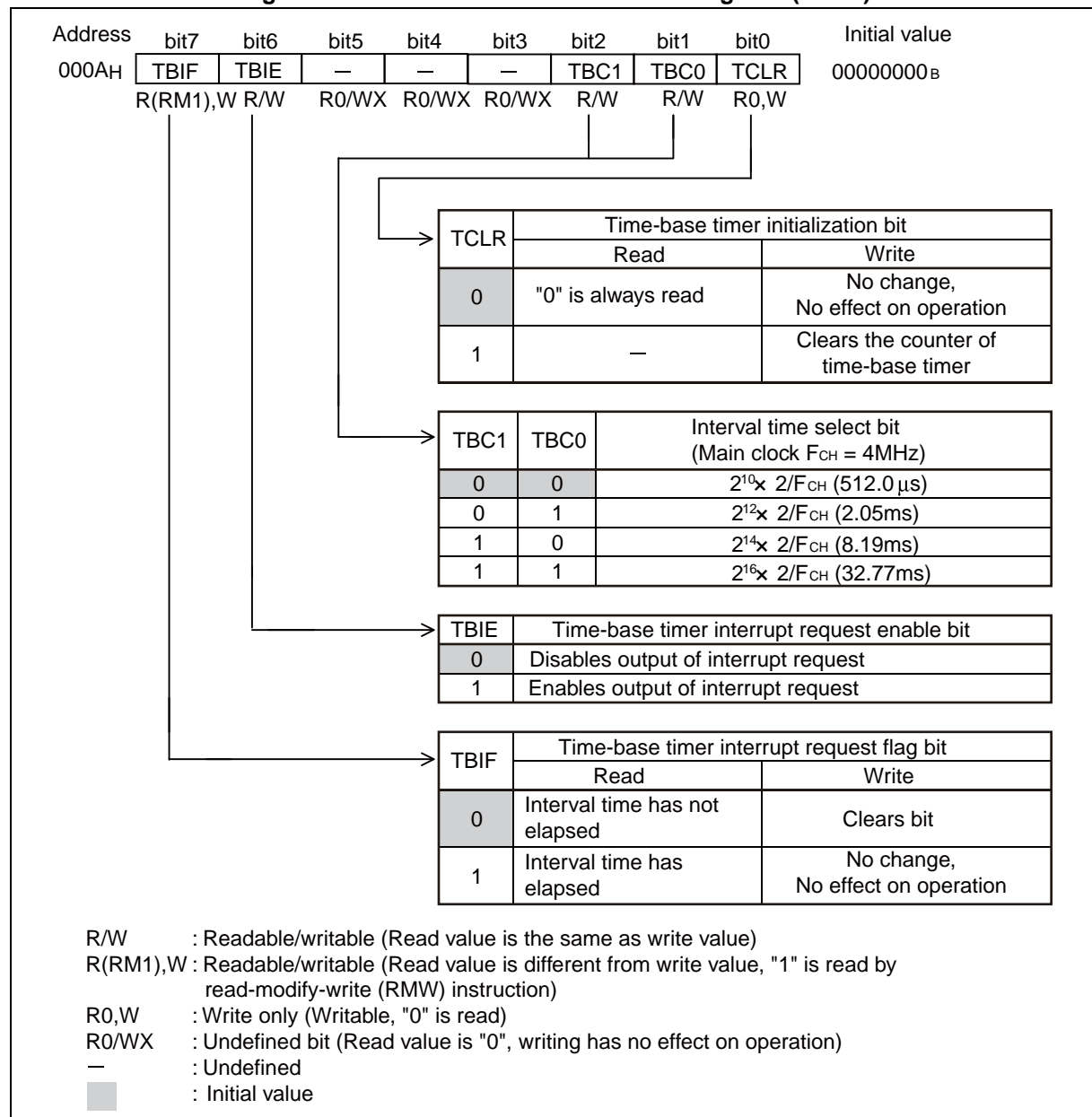


Table 10.3-1 Functional Description of Each Bit of Time-base Timer Control Register (TBTC)

Bit name		Function														
bit7	TBIF: Time-base timer interrupt request flag bit	Set to "1" when interval time selected by the time-base timer elapses. Interrupt request is outputted when this bit and the time-base timer interrupt request enable bit (TBIE) are set to "1". <b>Writing "0"</b> : clears the bit. <b>Writing "1"</b> : has no effect on operation. "1" is always read in read-modify-write (RMW) instruction.														
bit6	TBIE: Time-base timer interrupt request enable bit	This bit enables/disables output of interrupt requests to the interrupt controller. <b>Writing "0"</b> : disables output of time-base timer interrupt requests. <b>Writing "1"</b> : enables output of time-base timer interrupt requests. Interrupt request is outputted when this bit and the time-base timer interrupt request flag bit (TBIF) are set to "1".														
bit5 to bit3	Undefined bits	These bits are undefined. • The read value is always "0". • Writing has no effect on the operation.														
bit2, bit1	TBC1, TBC0: Interval time select bits	These bits select the interval time.														
		TBC1	TBC0	Interval time select bits (Main clock F <sub>CH</sub> = 4MHz)	0	0	2 <sup>10</sup> × 2/F <sub>CH</sub> (512.0 μs)	0	1	2 <sup>12</sup> × 2/F <sub>CH</sub> (2.05ms)	1	0	2 <sup>14</sup> × 2/F <sub>CH</sub> (8.19ms)	1	1	2 <sup>16</sup> × 2/F <sub>CH</sub> (32.77ms)
		TBC1	TBC0	Interval time select bits (Main clock F <sub>CH</sub> = 4MHz)												
		0	0	2 <sup>10</sup> × 2/F <sub>CH</sub> (512.0 μs)												
		0	1	2 <sup>12</sup> × 2/F <sub>CH</sub> (2.05ms)												
1	0	2 <sup>14</sup> × 2/F <sub>CH</sub> (8.19ms)														
1	1	2 <sup>16</sup> × 2/F <sub>CH</sub> (32.77ms)														
bit0	TCLR: Time-base timer initialization bit	This bit clears the time-base timer counter. <b>Writing "0"</b> : ignored and has no effect on the operation. <b>Writing "1"</b> : initializes all counter bits to "1". The read value is always "0". Note: When the output of the time-base timer is selected as the count clock for the watchdog timer, using this bit to clear the time-base timer also clears the watchdog timer.														

## 10.4 Interrupts of Time-base Timer

---

**An interrupt request is triggered when the interval time selected by the time-base timer elapses (interval timer function).**

---

### ■ Interrupt when Interval Function is in Operation

When the time-base timer counter counts down using the internal count clock and the selected time-base timer counter underflows, the time-base timer interrupt request flag bit (TBTC:TBIF) is set to "1". If the time-base timer interrupt request enable bit is enabled (TBTC:TBIE=1), an interrupt request (IRQ19) will be generated to interrupt controller.

- Regardless of the value of TBIE bit, TBIF bit is set to "1", when the selected bit underflows.
- When TBIF bit is set to "1" and TBIE bit is changed from the disable state to the enable state (0 → 1), an interrupt request is generated immediately.
- TBIF bit is not set when the counter is cleared (TBTC:TCLR = 1) and the time-base timer counter underflows at the same time.
- Write "1" to TBIF bit to clear an interrupt request in an interrupt processing routine.

---

**Note:**

When enabling the output of interrupt requests after canceling a reset (TBTC:TBIE = 1), always clear TBIF bit at the same time (TBTC:TBIF = 0).

---

**Table 10.4-1 Interrupts of Time-base Timer**

Item	Description
Interrupt condition	Interval time set by "TBTC:TBC1" and "TBC0" has elapsed
Interrupt flag	TBTC:TBIF
Interrupt enable	TBTC:TBIE

## ■ Register and Vector Table for Interrupts of Time-base Timer

Table 10.4-2 Register and Vector Table for Interrupts of Time-base Timer

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Time-base timer	IRQ19	ILR4	L19	FFD4 <sub>H</sub>	FFD5 <sub>H</sub>

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

## Note:

If the interval time set for the time-base timer is shorter than the main clock oscillation stabilization wait time, an interrupt request of the time-base timer is generated during the main clock oscillation wait time derived from the transition to the clock mode or standby mode. To prevent this, set the time-base timer interrupt request enable bit of the time-base timer control register (TBTC:TBIE) to "0" to disable interrupts of the time-base timer when entering a mode in which the main clock stops oscillating (stop mode, sub clock mode or sub PLL clock mode).

## 10.5 Explanation of Time-base Timer Operations and Setup Procedure Example

This section describes the operations of the interval timer function of the time-base timer.

### ■ Operations of Time-base Timer

The counter of the time-base timer is initialized to "3FFFFFF<sub>H</sub>" after a reset and starts counting while being synchronized with the main clock divided by two.

The time-base timer continues to count down as long as the main clock is oscillating. Once the main clock halts, the counter stops counting and is initialized to "3FFFFFF<sub>H</sub>".

The settings shown in Figure 10.5-1 are required to use the interval timer function.

**Figure 10.5-1 Settings of Interval Timer Function**

TBTC		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Address: 000A <sub>H</sub>		TBIF	TBIE	-	-	-	TBC1	TBC0	TCLR
		0	1				⊙	⊙	0

⊙: Bit used  
 1: Set to "1"  
 0: Set to "0"

When the time-base timer initialization bit in the time-base timer control register (TBTC:TCLR) is set to "1", the counter of the time-base timer is initialized to "3FFFFFF<sub>H</sub>" and continues to count down. When the selected interval time has elapsed, the time-base timer interrupt request flag bit of the time-base timer control register (TBTC:TBIF) becomes "1". In other words, an interrupt request is generated at each interval time selected, based on the time when the counter was last cleared.

### ■ Clearing Time-base Timer

If the time-base timer is cleared when the output of the time-base timer is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

When clearing the counter by using the time-base timer initialization bit (TBTC:TCLR), perform setup so that this does not have unexpected effects on other peripheral functions.

When the output of the time-base timer is selected as the count clock for the watchdog timer, clearing the time-base timer also clears the watchdog timer.

The time-base timer is cleared not only by the time-base timer initialization bit (TBTC:TCLR), but also when the main clock is stopped and a count is required for the oscillation stabilization wait time. More specifically, the time-base timer is cleared in the following situations:

- When moving from the main clock mode or main PLL clock mode to the stop mode
- When moving from the main clock mode or main PLL clock mode to the sub clock mode or sub PLL clock mode
- At power on
- At low-voltage detection reset

The counter of the time-base timer is also cleared and stops the operation if a reset occurs while the main clock is still running after the main clock oscillation stabilization wait time has elapsed. The counter, however, continues to operate during a reset if a count is required for the oscillation stabilization wait time.

### ■ Operating Examples of Time-base Timer

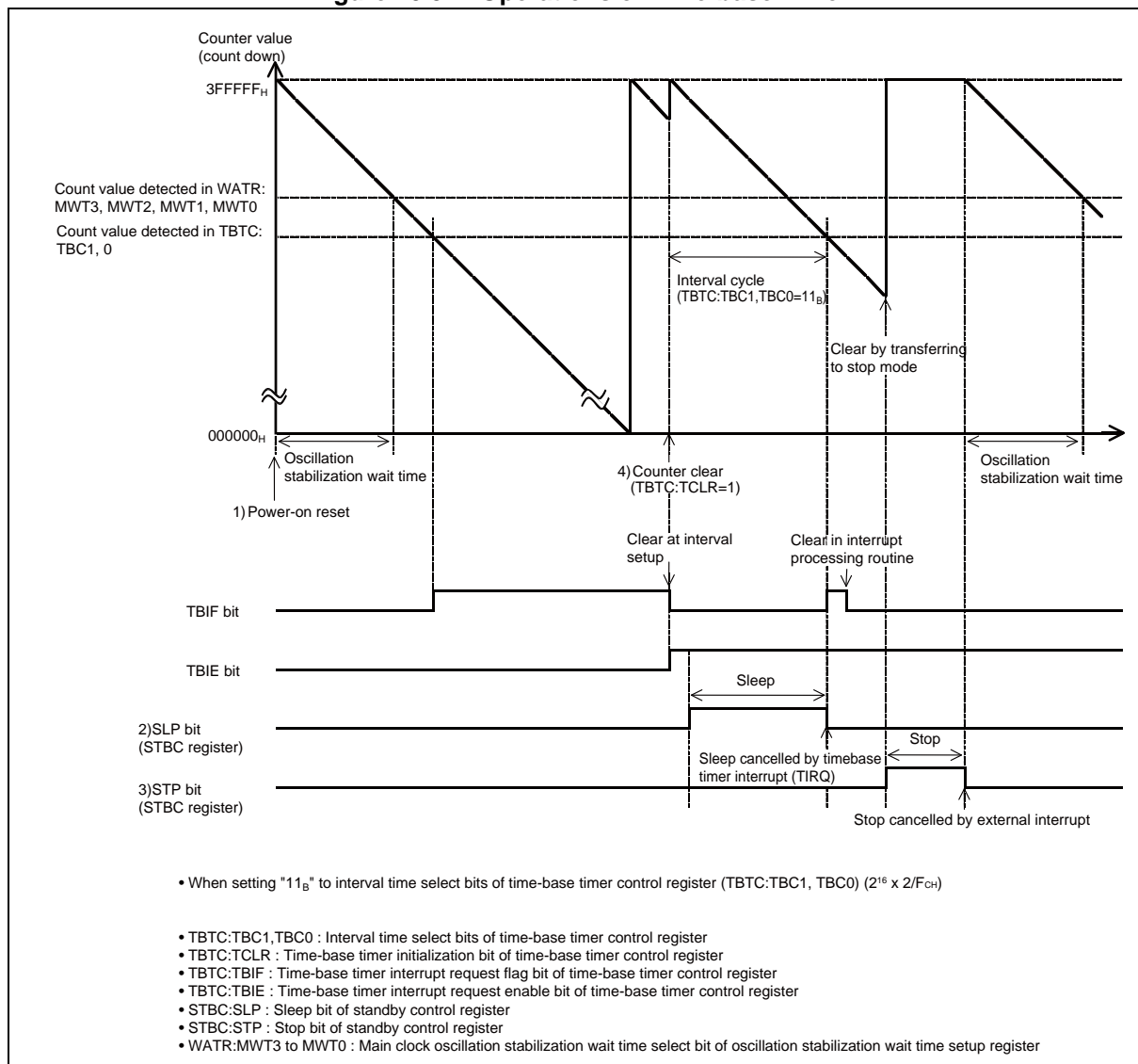
Figure 10.5-2 shows operating examples of operation under the following conditions:

- 1) When a power-on reset is generated
- 2) When entering the sleep mode during the operation of the interval timer function in the main clock mode or main PLL clock mode
- 3) When entering the stop mode during the main clock mode or main PLL clock mode
- 4) When a request is issued to clear the counter

The same operation is performed when changing to the time-base timer mode as for when changing to the sleep mode.

In the sub clock mode, sub PLL clock mode, main clock mode and main PLL clock mode, the timer operation is stopped during the stop mode, as the time-base timer is cleared and the main clock halts. Upon recovering from the stop mode, the time-base timer is used to count the oscillation stabilization wait time.

Figure 10.5-2 Operations of Time-base Timer



## ■ Setup Procedure Example

### ● Initial setting

The time-base timer is set up in the following procedure:

- 1) Disable interrupts. (TBTC:TBIE = 0)
- 2) Set the interval time. (TBTC:TBC1, TBC0)
- 3) Enable interrupts. (TBTC:TBIE = 1)
- 4) Clear the counter. (TBTC:TCLR = 1)

### ● Interrupt processing

- 1) Clear the interrupt request flag. (TBTC:TBIF = 0)
- 2) Clear the counter. (TBTC:TCLR = 1)

## 10.6 Notes on Using Time-base Timer

---

Care must be taken for the following points when using the Time-base Timer.

---

### ■ Notes on Using Time-base Timer

- When setting the timer by program

The timer cannot be recovered from interrupt processing, when the time-base timer interrupt request flag bit (TBTC:TBIF) is set to "1" and the interrupt request enable bit is enabled (TBTC:TBIE = 1). Always clear TBIF bit in the interrupt processing routine.

- Clearing time-base timer

The time-base timer is cleared not only by the time-base timer initialization bit (TBTC:TCLR=1) but also when the oscillation stabilization wait time is required for the main clock. When the time-base timer is selected for the count clock of the watchdog timer (WDTC:CS1, CS0 = 00<sub>B</sub> or CS1, CS0 = 01<sub>B</sub>), clearing the time-base timer also clears the watchdog timer.

- Peripheral functions receiving clock from time-base timer

In the mode where the source oscillation of the main clock is stopped, the counter is cleared and the time-base timer stops operation. In addition, if the time-base timer is cleared when the output of the time-base timer is used in other peripheral functions, this will affect the operation such as cycle change. The clock for the watchdog timer is also outputted from the initial state. However, as the watchdog timer counter is cleared at the same time, the watchdog timer operates in the normal cycles.





# ***CHAPTER 11***

---

# ***WATCHDOG TIMER***

**This chapter describes the functions and operations of the watchdog timer.**

- 11.1 Overview of Watchdog Timer
- 11.2 Configuration of Watchdog Timer
- 11.3 Register of The Watchdog Timer
- 11.4 Explanation of Watchdog Timer Operations and Setup Procedure Example
- 11.5 Notes on Using Watchdog Timer

## 11.1 Overview of Watchdog Timer

The watchdog timer functions as a counter used to prevent programs from running out of control.

### ■ Watchdog Timer Function

The watchdog timer functions as a counter used to prevent programs from running out of control. Once the watchdog timer is activated, its counter needs to be cleared at specified intervals regularly. A watchdog reset is generated if the timer is not cleared within a certain amount of time due to a problem such as the program entering an infinite loop.

The output of either the time-base timer or watch prescaler can be selected as the count clock for the watchdog timer.

The interval times of the watchdog timer are shown in Table 11.1-1. If the counter of the watchdog timer is not cleared, a watchdog reset is generated between the minimum time and the maximum time. Clear the counter of the watchdog timer within the minimum time.

**Table 11.1-1 Interval Times of Watchdog Timer**

Count clock type	Count clock switch bits (WDTC:CS1, CS0)*	Interval time	
		Minimum time	Maximum time
Time-base timer output (main clock = 4MHz)	00 <sub>B</sub>	524 ms	1.05 s
	01 <sub>B</sub>	262 ms	524 ms
Watch prescaler output (sub clock = 32.768kHz)	10 <sub>B</sub>	500 ms	1.00 s
	11 <sub>B</sub>	250 ms	500 ms

\*: WDTC:CS1, 0: Count clock switch bit of watchdog timer control register

For information about the minimum and maximum times of the watchdog timer interval, refer to "11.4 Explanation of Watchdog Timer Operations and Setup Procedure Example".

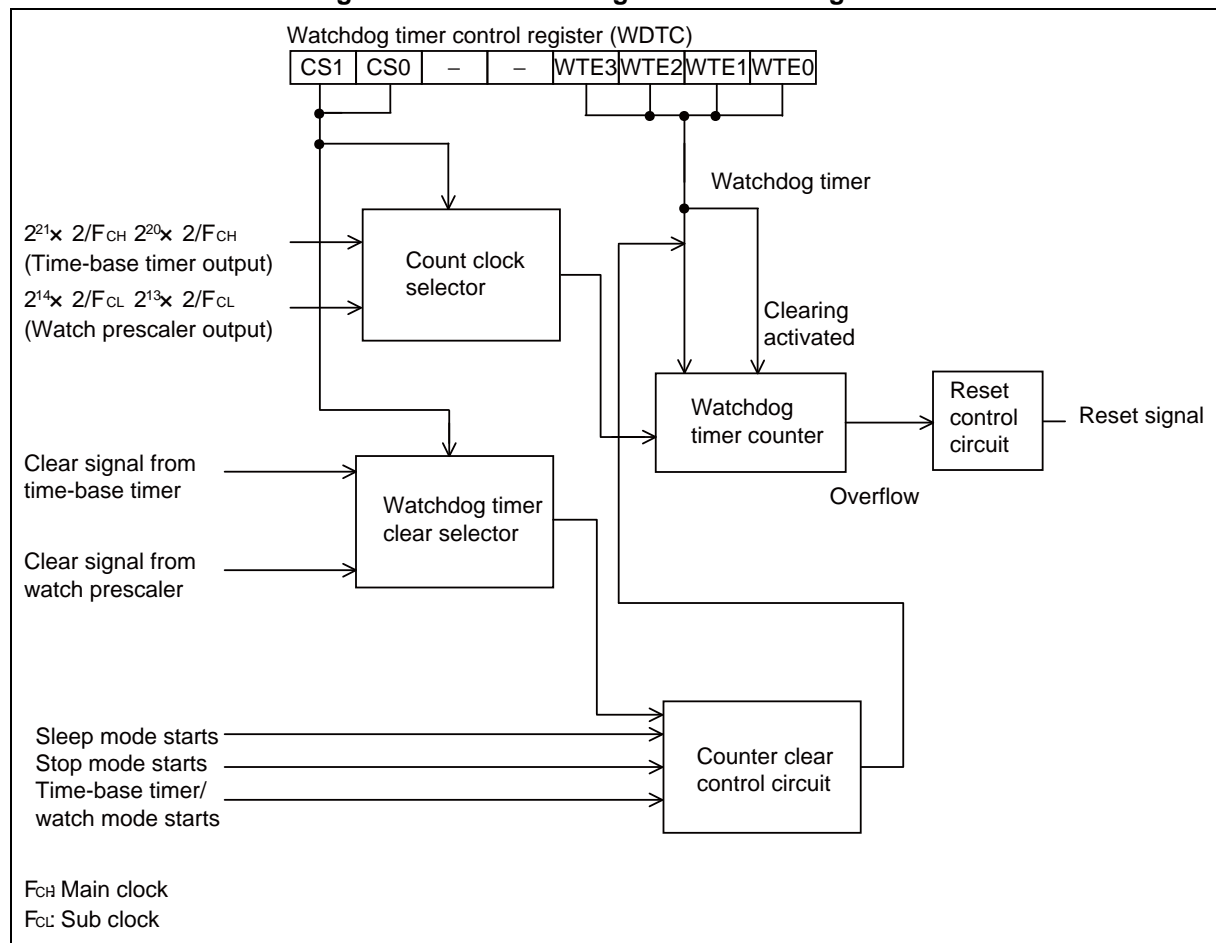
## 11.2 Configuration of Watchdog Timer

The watchdog timer consists of the following blocks:

- Count clock selector
- Watchdog timer counter
- Reset control circuit
- Watchdog timer clear selector
- Counter clear control circuit
- Watchdog Timer Control Register (WDTC)

### ■ Block Diagram of Watchdog Timer

Figure 11.2-1 Block Diagram of Watchdog Timer



- Count clock selector

This selector selects the count clock of the watchdog timer counter.

- Watchdog timer counter

This is a 1-bit counter that uses the output of either the time-base timer or watch prescaler as the count clock.

- Reset control circuit

This circuit generates a reset signal when the watchdog timer counter overflows.

- Watchdog timer clear selector

This selector selects the watchdog timer clear signal.

- Counter clear control circuit

This circuit controls the clearing and stopping of the watchdog timer counter.

- Watchdog timer control register (WDTC)

This register performs setup for activating/clearing the watchdog timer counter as well as for selecting the count clock.

## ■ Input Clock

The watchdog timer uses the output clock from either the time-base timer or watch prescaler as the input clock (count clock).

## 11.3 Register of The Watchdog Timer

Figure 11.3-1 shows the register of the watchdog timer.

### ■ Register of The Watchdog Timer

Figure 11.3-1 Register of The Watchdog Timer

Watchdog timer control register (WDTC)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
000C <sub>H</sub>	CS1	CS0	–	–	WTE3	WTE2	WTE1	WTE0	00000000 <sub>B</sub>
	R/W	R/W	R0/WX	R0/WX	R0,W	R0,W	R0,W	R0,W	

R/W : Readable/writable (Read value is the same as write value)  
R0,W : Write only (Writable, "0" is read)  
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)  
– : Undefined

## 11.3.1 Watchdog Timer Control Register (WDTC)

The watchdog timer control register (WDTC) activates or clears the watchdog timer.

### ■ Watchdog Timer Control Register (WDTC)

Figure 11.3-2 Watchdog Timer Control Register (WDTC)

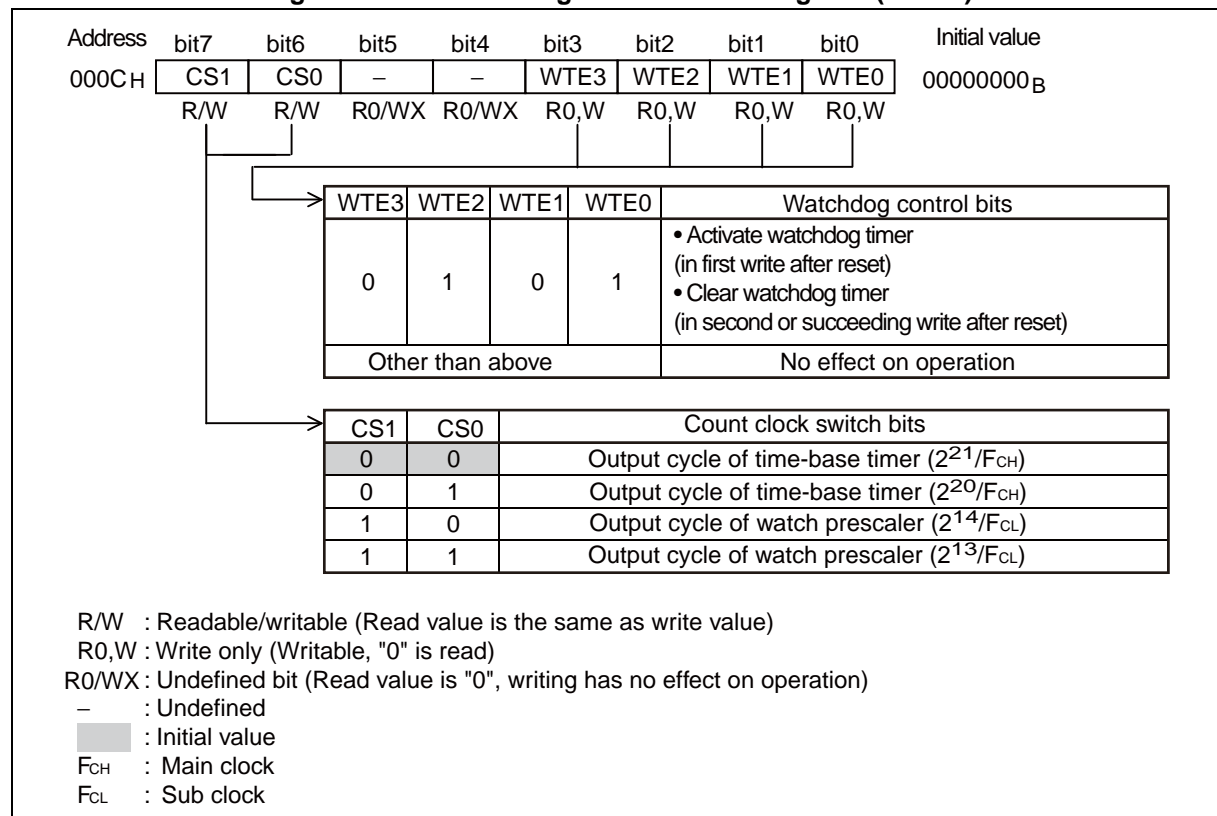


Table 11.3-1 Functional Description of Each Bit of Watchdog Timer Control Register (WDTC)

Bit name		Function															
bit7, bit6	CS1, CS0: Count clock switch bits	These bits select the count clock of the watchdog timer.															
		<table><tr><th>CS1</th><th>CS0</th><th>Count clock switch bits</th></tr><tr><td>0</td><td>0</td><td>Output cycle of time-base timer (<math>2^{21}/F_{CH}</math>)</td></tr><tr><td>0</td><td>1</td><td>Output cycle of time-base timer (<math>2^{20}/F_{CH}</math>)</td></tr><tr><td>1</td><td>0</td><td>Output cycle of watch prescaler (<math>2^{14}/F_{CL}</math>)</td></tr><tr><td>1</td><td>1</td><td>Output cycle of watch prescaler (<math>2^{13}/F_{CL}</math>)</td></tr></table>	CS1	CS0	Count clock switch bits	0	0	Output cycle of time-base timer ( $2^{21}/F_{CH}$ )	0	1	Output cycle of time-base timer ( $2^{20}/F_{CH}$ )	1	0	Output cycle of watch prescaler ( $2^{14}/F_{CL}$ )	1	1	Output cycle of watch prescaler ( $2^{13}/F_{CL}$ )
		CS1	CS0	Count clock switch bits													
		0	0	Output cycle of time-base timer ( $2^{21}/F_{CH}$ )													
		0	1	Output cycle of time-base timer ( $2^{20}/F_{CH}$ )													
		1	0	Output cycle of watch prescaler ( $2^{14}/F_{CL}$ )													
		1	1	Output cycle of watch prescaler ( $2^{13}/F_{CL}$ )													
<ul style="list-style-type: none"><li>Write to these bits at the same time as activating the watchdog timer by the watchdog control bits.</li><li>No change can be made once the watchdog timer is activated.</li></ul>																	
Note: Always select the output of the watch prescaler in the sub clock mode or sub PLL clock mode, as the time-base timer is stopped in these modes. Do not select the output of the watch prescaler in single clock product.																	
bit5, bit4	Undefined bits	<p>These bits are undefined.</p> <ul style="list-style-type: none"><li>The read value is "00<sub>B</sub>".</li><li>Writing has no effect on the operation.</li></ul>															
bit3 to bit0	WTE3, WTE2, WTE1, WTE0: Watchdog control bits	<p>These bits are used to control the watchdog timer.</p> <p><b>Writing "0101<sub>B</sub>"</b>: activates the watchdog timer (in first write after reset) or clears it (in second or succeeding write after reset).</p> <p><b>Writing other than "0101<sub>B</sub>"</b>: has no effect on operation.</p> <ul style="list-style-type: none"><li>The read value is "0000<sub>B</sub>".</li></ul>															

Read-modify-write (RMW) instructions cannot be used.



## 11.4 Explanation of Watchdog Timer Operations and Setup Procedure Example

---

**The watchdog timer generates a watchdog reset when the watchdog timer counter overflows.**

---

### ■ Operations of Watchdog Timer

#### ● How to activate the watchdog timer

- The timer of the watchdog timer is activated when "0101<sub>B</sub>" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0) for the first time after a reset. The count clock switch bits of the watchdog timer control register (WDTC:CS1,CS0) should also be set at the same time.
- Once the watchdog timer is activated, a reset is the only way to stop its operation.

#### ● Clearing the watchdog timer

- When the counter of the watchdog timer is not cleared within the interval time, it overflows, allowing the watchdog timer to generate a watchdog reset.
- The counter of the watchdog timer is cleared when "0101<sub>B</sub>" is written to the watchdog control bits of the watchdog timer control register (WDTC:WTE3 to WTE0) for the second or any succeeding time.
- The watchdog timer is cleared at the same time as the timer selected as the count clock (time-base timer or watch prescaler) is cleared.

#### ● Operations in standby mode

Regardless of the clock mode selected, the watchdog timer clears its counter and stops the operation when entering a standby mode (sleep/stop/time-base timer/watch).

Once released from the standby mode, the timer restarts the operation.

---

#### Note:

The watchdog timer is also cleared when the timer selected as the count clock (time-base timer or watch prescaler) is cleared.

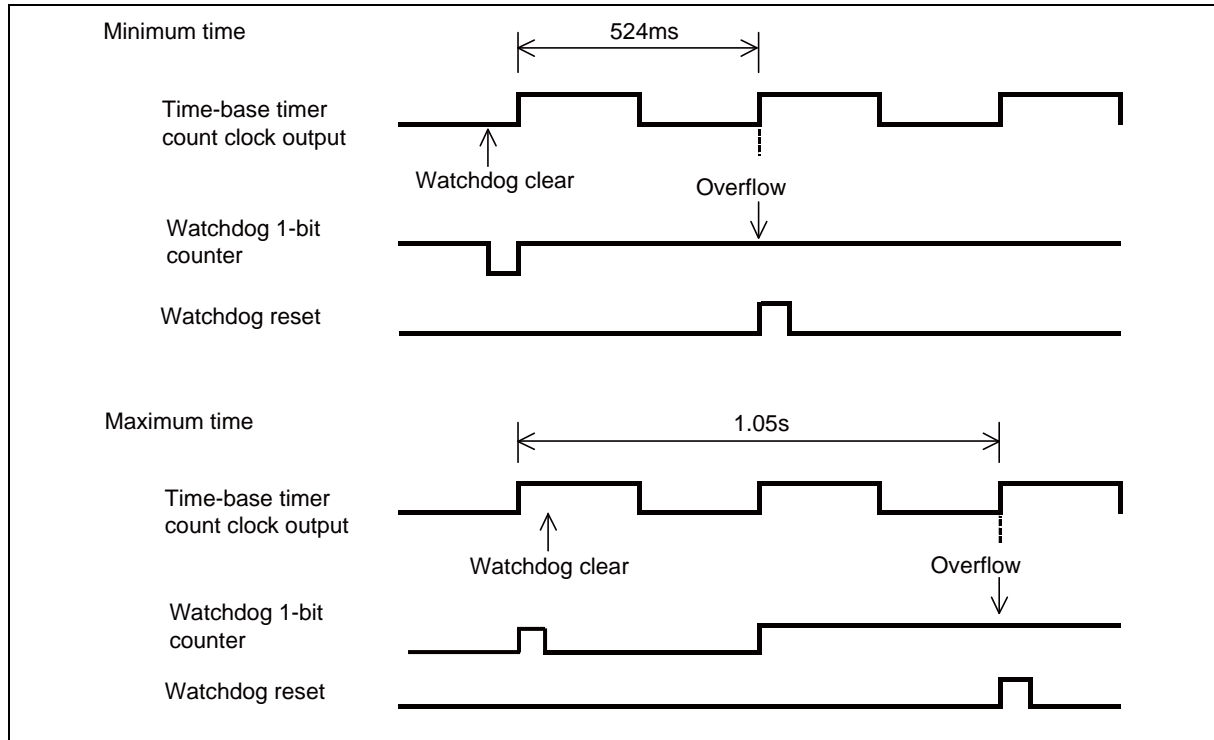
For this reason, the watchdog timer cannot function as such, if the software is set to clear the selected timer repeatedly during the interval time of the watchdog timer.

---

● Interval time

The interval time varies depending on the timing for clearing the watchdog timer. Figure 11.4-1 shows the correlation between the clearing timing of the watchdog timer and the interval time.

**Figure 11.4-1 Clearing Timing and Interval Time of Watchdog Timer**  
(Main clock = 4MHz, WDTC:CS1, CS0=00<sub>B</sub>)



● Operation in the sub clock mode

When a watchdog reset is generated in the sub clock mode, the timer starts operating in the main clock mode after the oscillation stabilization wait time has elapsed. The reset signal is outputted during this oscillation stabilization wait time.

■ Setup Procedure Example

The watchdog timer is set up in the following procedure:

- 1) Select the count clock. (WDTC:CS1, CS0)
- 2) Activate the watchdog timer. (WDTC:WTE3 to WTE0 = 0101<sub>B</sub>)
- 3) Clear the watchdog timer. (WDTC:WTE3 to WTE0 = 0101<sub>B</sub>)

## 11.5 Notes on Using Watchdog Timer

---

**Care must be taken for the following points when using the watchdog timer.**

---

### ■ Notes on Using Watchdog Timer

- Stopping the watchdog timer

Once activated, the watchdog timer cannot be stopped until a reset is generated.

- Selecting the count clock

The count clock switch bits (WDTC:CS1, 0) can be rewritten only when the watchdog control bits (WDTC:WTE3 to WTE0) are set to "0101<sub>B</sub>" upon the activation of the watchdog timer. The count clock switch bits cannot be written by a bit operation instruction. Moreover, the bit settings should not be changed once the timer is activated.

In the sub clock mode, the time-base timer does not operate because the main clock stops oscillating.

In order to operate the watchdog timer in the sub clock mode, it is necessary to select the watch prescaler as the count clock beforehand and set "WDTC:CS1, 0" to "10<sub>B</sub>" or "11<sub>B</sub>".

- Clearing the watchdog timer

Clearing the counter used for the count clock of the watchdog timer (time-base timer or watch prescaler) also clears the counter of the watchdog timer.

The counter of the watchdog timer is cleared when entering the sleep mode, stop mode or watch mode.

- Programming precaution

When creating a program in which the watchdog timer is cleared repeatedly in the main loop, set the processing time of the main loop including the interrupt processing time to the minimum watchdog timer interval time or shorter.

# ***CHAPTER 12***

---

# ***WATCH PRESCALER***

**This chapter describes the functions and operations of the watch prescaler.**

- 12.1 Overview of Watch Prescaler
- 12.2 Configuration of Watch Prescaler
- 12.3 Registers of the Watch Prescaler
- 12.4 Interrupts of Watch Prescaler
- 12.5 Explanation of Watch Prescaler Operations and Setup Procedure Example
- 12.6 Notes on Using Watch Prescaler
- 12.7 Sample Programs for Watch Prescaler

## 12.1 Overview of Watch Prescaler

The watch prescaler is a 15-bit down-counting, free-run counter, which is synchronized with the sub clock divided by two. It has an interval timer function that continuously generates interrupt requests at regular intervals.

### ■ Interval Timer Function

The interval timer function continuously generates interrupt requests at regular intervals, using the sub clock divided by two as its count clock.

- The counter of the watch prescaler counts down and an interrupt request is generated every time the selected interval time has elapsed.
- The interval time can be selected from the following four types:

Table 12.1-1 shows the interval times of the watch prescaler.

**Table 12.1-1 Interval Times of Watch Prescaler**

Internal count clock cycle	Interval time
$2/F_{CL}$ (61.0 $\mu$ s)	$2^{11} \times 2/F_{CL}$ (125ms)
	$2^{12} \times 2/F_{CL}$ (250ms)
	$2^{13} \times 2/F_{CL}$ (500ms)
	$2^{14} \times 2/F_{CL}$ (1.00s)

$F_{CL}$ : sub clock

The values in parentheses represent the values achieved when the sub clock operates at 32.768kHz.

Note:

The watch prescaler cannot be used in single clock product.



- Watch prescaler counter (counter)

This is a 15-bit down-counter that uses the sub clock divided by two as its count clock.

- Counter clear circuit

This circuit controls the clearing of the watch prescaler.

- Interval timer selector

This circuit selects one out of the four bits used for the interval timer among 15 bits available in the watch prescaler counter.

- Watch prescaler control register (WPCR)

This register selects the interval time, clears the counter, controls interrupts and checks the status.

## ■ Input Clock

The watch prescaler uses the sub clock divided by two as its input clock (count clock).

## ■ Output Clock

The watch prescaler supplies its clock to the timer for the oscillation stabilization wait time of the sub clock, the watchdog timer and the watch counter.

## 12.3 Registers of the Watch Prescaler

Figure 12.3-1 shows the register of the watch prescaler.

### ■ Register of the Watch Prescaler

**Figure 12.3-1 Register of the Watch Prescaler**

Watch Prescaler Control Register (WPCR)									Initial value
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000B <sub>H</sub>	WTIF	WTIE	–	–	–	WTC1	WTC0	WCLR	00000000 <sub>B</sub>
	R(RM1),W	R/W	R0/WX	R0/WX	R0/WX	R/W	R/W	R0,W	

R/W : Readable/writable (Read value is the same as write value)  
 R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)  
 R0,W : Write only (Writable, "0" is read)  
 R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)  
 – : Undefined



## 12.3.1 Watch Prescaler Control Register (WPCR)

The watch prescaler control register (WPCR) is a register used to select the interval time, clear the counter, control interrupts and check the status.

### ■ Watch Prescaler Control Register (WPCR)

Figure 12.3-2 Watch Prescaler Control Register (WPCR)

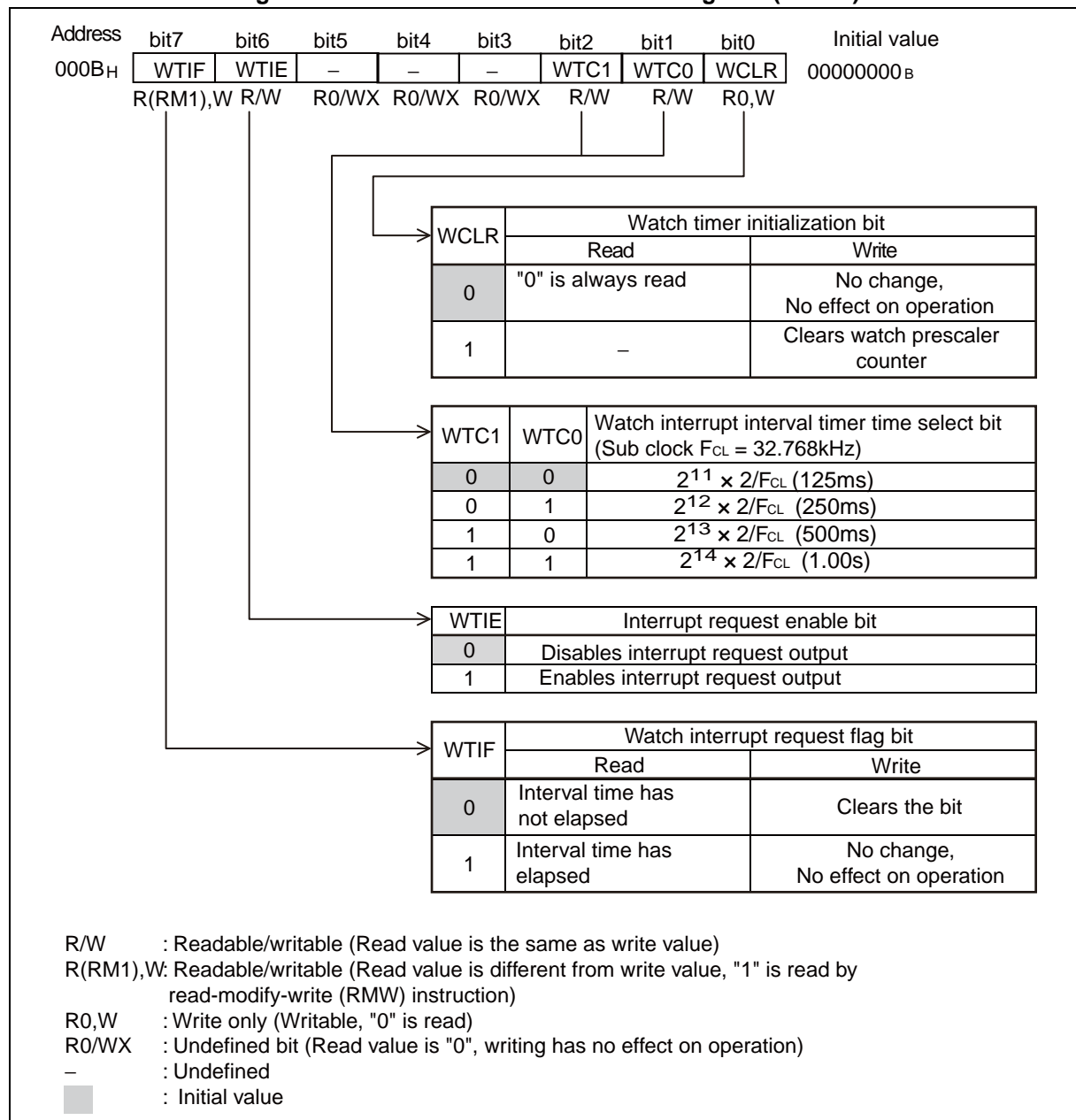


Table 12.3-1 Functional Description of Each Bit of Watch Prescaler Control Register (WPCR)

Bit name		Function															
bit7	WTIF: Watch interrupt request flag bit	<p>This bit becomes "1" when the selected interval time of the watch prescaler has elapsed.</p> <ul style="list-style-type: none"> <li>Interrupt requests are generated when this bit and the interrupt request enable bit (WTIE) are set to "1".</li> </ul> <p><b>Writing "0"</b>: sets this bit to "0".</p> <p><b>Writing "1"</b>: ignored and has no effect on the operation.</p> <ul style="list-style-type: none"> <li>"1" is always read in read-modify-write (RMW) instruction.</li> </ul>															
bit6	WTIE: Interrupt request enable bit	<p>This bit enables/disables output of interrupt requests to the interrupt controller.</p> <p><b>Writing "0"</b>: disables the interrupt request output of the watch prescaler.</p> <p><b>Writing "1"</b>: enables the interrupt request output of the watch prescaler.</p> <p>Interrupt requests are outputted when this bit and the watch interrupt request flag bit (WTIF) are set to "1".</p>															
bit5 to bit3	Undefined bits	<p>These bits are undefined.</p> <ul style="list-style-type: none"> <li>The read value is always "0".</li> <li>Writing has no effect on the operation.</li> </ul>															
bit2, bit1	WTC1, WTC0: Watch interrupt interval time select bits	<p>These bits select the interval time.</p> <table border="1"> <thead> <tr> <th>WTC1</th><th>WTC0</th><th>Interval time select bits (sub clock <math>F_{CL} = 32.768\text{kHz}</math>)</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td><math>2^{11} \times 2/F_{CL}(125\text{ms})</math></td></tr> <tr> <td>0</td><td>1</td><td><math>2^{12} \times 2/F_{CL}(250\text{ms})</math></td></tr> <tr> <td>1</td><td>0</td><td><math>2^{13} \times 2/F_{CL}(500\text{ms})</math></td></tr> <tr> <td>1</td><td>1</td><td><math>2^{14} \times 2/F_{CL}(1.00\text{s})</math></td></tr> </tbody> </table>	WTC1	WTC0	Interval time select bits (sub clock $F_{CL} = 32.768\text{kHz}$ )	0	0	$2^{11} \times 2/F_{CL}(125\text{ms})$	0	1	$2^{12} \times 2/F_{CL}(250\text{ms})$	1	0	$2^{13} \times 2/F_{CL}(500\text{ms})$	1	1	$2^{14} \times 2/F_{CL}(1.00\text{s})$
WTC1	WTC0	Interval time select bits (sub clock $F_{CL} = 32.768\text{kHz}$ )															
0	0	$2^{11} \times 2/F_{CL}(125\text{ms})$															
0	1	$2^{12} \times 2/F_{CL}(250\text{ms})$															
1	0	$2^{13} \times 2/F_{CL}(500\text{ms})$															
1	1	$2^{14} \times 2/F_{CL}(1.00\text{s})$															
bit0	WCLR: Watch timer initialization bit	<p>This bit clears the counter for the watch prescaler.</p> <p><b>Writing "0"</b>: ignored and has no effect on the operation.</p> <p><b>Writing "1"</b>: initializes all counter bits to "1".</p> <p>The read value is always "0".</p> <p>Note: When the output of the watch prescaler is selected as the count clock of the watchdog timer, clearing the watch prescaler with this bit also clears the watchdog timer.</p>															

## 12.4 Interrupts of Watch Prescaler

**An interrupt request is generated when the selected interval time of the watch prescaler has elapsed (interval timer function).**

### ■ Interrupts in Operation of Interval Timer Function (Watch Interrupts)

In any mode other than the main clock stop mode, the watch interrupt request flag bit is set to "1" (WPCR:WTIF = 1), when the watch prescaler counter counts up by using the source oscillation of the sub clock and the time of the interval timer has elapsed. If the interrupt request enable bit is also enabled (WPCR:WTIE = 1) and watch counter start interrupt request enable bit of the watch counter is disabled (WCSR:ISEL=0), an interrupt request (IRQ20) occurs from watch prescaler to an interrupt controller.

- Regardless of the value in the WTIE bit, the WTIF bit is set to "1" when the time set by the watch interrupt interval time select bits has been reached.
- When the WTIF bit is set to "1", changing the WTIE bit from the disable state to the enable state (WPCR:WTIE = 0 → 1) immediately generates an interrupt request.
- The WTIF bit cannot be set when the counter is cleared (WPCR:WCLR = 1) at the same time as the selected bit overflows.
- Write "0" to the WTIF bit in the interrupt processing routine to clear an interrupt request to "0".

Note:

When enabling the output of interrupt requests (WPCR:WTIE = 1) after canceling a reset, always clear the WTIF bit at the same time (WPCR:WTIF=0).

### ■ Interrupts of Watch Prescaler

**Table 12.4-1 Interrupts of Watch Prescaler**

Item	Description
Interrupt condition	Interval time set by "WPCR:WTC1" and "WTC0" has elapsed.
Interrupt flag	WPCR:WTIF
Interrupt enable	WPCR:WTIE

**■ Register and Vector Table Related to Interrupts of Watch Prescaler**
**Table 12.4-2 Register and Vector Table Related to Interrupts of Watch Prescaler**

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Watch prescaler*	IRQ20	ILR5	L20	FFD2 <sub>H</sub>	FFD3 <sub>H</sub>

\*: The watch prescaler shares the same interrupt request number and vector table as the watch counter.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

**Note:**

If the interval time set for the watch prescaler is shorter than the oscillation stabilization wait time of the sub clock, an interrupt request of the watch prescaler is generated during the oscillation stabilization wait time of the sub clock required for recovery by an external interrupt upon the transition from the sub clock mode or the sub PLL clock mode to the stop mode. To prevent this, set the interrupt request enable bit (WPCR:WTIE) in the watch prescaler control register to "0" to disable interrupts of the watch prescaler when entering the stop mode during the sub clock mode or the sub PLL clock mode.

## 12.5 Explanation of Watch Prescaler Operations and Setup Procedure Example

---

The watch prescaler operates as an interval timer.

---

### ■ Operations of Interval Timer Function (Watch Prescaler)

The counter of the watch prescaler continues to count down using the sub clock divided by two as its count clock as long as the sub clock oscillates.

When cleared (WPCR:WCLR=1), the counter starts to count down from "7FFF<sub>H</sub>". Once "0000<sub>H</sub>" is reached, the counter returns to "7FFF<sub>H</sub>" to continue the count. When the time set by the interrupt interval time select bits is reached during down-counting, the watch interrupt request flag bit (WPCR:WTIF) is set to "1" in any mode other than the main clock stop mode. In other words, a watch interrupt request is generated at each selected interval time, based on the time when the counter was last cleared.

### ■ Clearing Watch Prescaler

If the watch prescaler is cleared when the output of the watch prescaler is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

When clearing the counter by using the watch prescaler initialization bit (WPCR:WCLR), perform setup so that this does not have unexpected effects on other peripheral functions.

When the output of the watch prescaler is selected as the count clock, clearing the watch prescaler also clears the watchdog timer.

The watch prescaler is cleared not only by the watch prescaler initialization bit (WPCR:WCLR) but also when the sub clock is stopped and a count is required for the oscillation stabilization wait time.

- When moving from the sub clock mode or sub PLL clock mode to the stop mode
- When the sub clock oscillation stop bit in the system clock control register (SYCC:SUSBS) is set to "1" in the main clock mode or main PLL clock mode

In addition, the counter of the watch prescaler is cleared and stops operation when a reset is generated.

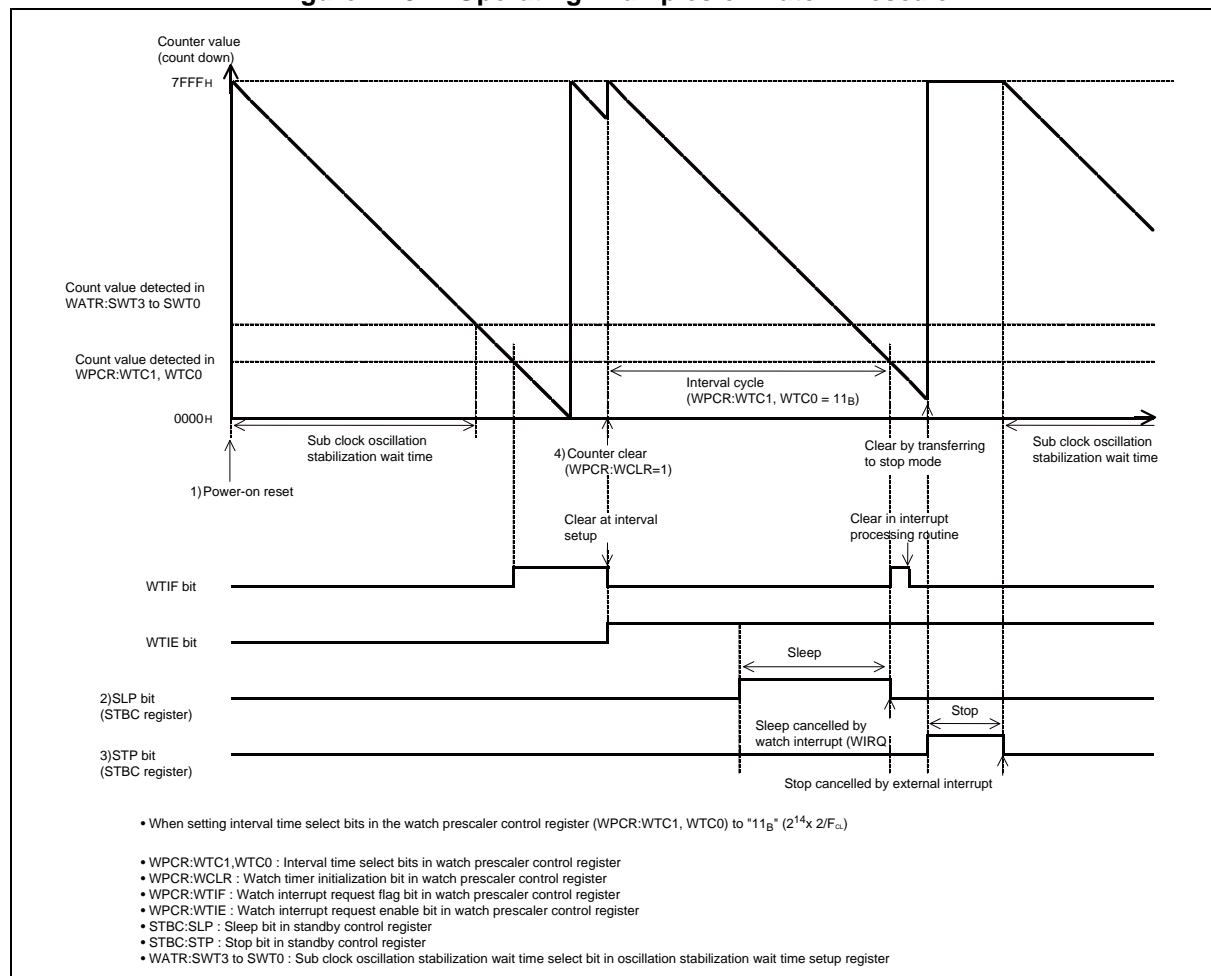
### ■ Operating Examples of Watch Prescaler

Figure 12.5-1 shows operating examples under the following conditions:

- 1) When a power-on reset is generated
- 2) When entering the sleep mode during the operation of the interval timer function in the sub clock mode or sub PLL clock mode
- 3) When entering the stop mode during the operation of the interval timer function in the sub clock mode or sub PLL clock mode
- 4) When a request is issued to clear the counter

The same operation is performed when changing to the watch mode as for when changing to the sleep mode.

Figure 12.5-1 Operating Examples of Watch Prescaler



## ■ Setup Procedure Example

The watch prescaler is set up in the following procedure:

### ● Initial setting

- 1) Set the interrupt level. (ILR5)
- 2) Set the interval time. (WPCR:WTC1, WTC0)
- 3) Enable interrupts. (WPCR:WTIE = 1)
- 4) Clear the counter. (WPCR:WCLR = 1)

### ● Interrupt processing

- 1) Clear the interrupt request flag. (WPCR:WTIF = 0)
- 2) Arbitrary processing

## 12.6 Notes on Using Watch Prescaler

---

**Shown below are the precautions that must be followed when using the watch prescaler.**

**The watch prescaler cannot be used in single clock option product.**

---

### ■ Notes on Using Watch Prescaler

- When setting the prescaler by program

The prescaler cannot be recovered from interrupt processing when the watch interrupt request flag bit (WPCR:WTIF) is set to "1" and the interrupt request enable bit is enabled (WPCR:WTIE = 1). Always clear the WTIF bit within the interrupt routine.

- Clearing the watch prescaler

When the watch prescaler is selected as the count clock of the watchdog timer (WDTC:CS1, CS0 = 10<sub>B</sub> or CS1, CS0 = 11<sub>B</sub>), clearing the watch prescaler also clears the watchdog timer.

- Watch interrupts

In the main clock stop mode, the watch prescaler performs counting but does not generate the watch prescaler interrupts (IRQ20).

- Peripheral functions receiving clock from the watch prescaler

If the watch prescaler is cleared when the output of the watch prescaler is used in other peripheral functions, this will affect the operation by changing the count time or in other manners.

The clock for the watchdog timer is also outputted from the initial state. However, as the watchdog timer counter is cleared at the same time as the prescaler counter, the watchdog timer operates in the normal cycles.

## 12.7 Sample Programs for Watch Prescaler

We provide sample programs that can be used to operate the watch prescaler.

### ■ Sample Programs for Watch Prescaler

For information about sample programs for the watch prescaler, refer to "■ Sample Programs" in Preface.

### ■ Setting Methods not Covered by Sample Programs

#### ● How to initialize the watch prescaler

The watch timer initialization bit (WPCR:WCLR) is used.

Control item	Watch timer initialization bit (WCLR)
When initializing watch prescaler	Set the bit to "1".

#### ● How to select the interval time

The watch interrupt interval time select bits (WPCR:WTC1/WTC0) are used to select the interval time.

#### ● Interrupt-related register

The interrupt level is set using the interrupt level register shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
Watch prescaler	Interrupt level register (ILR5) Address: 0007E <sub>H</sub>	#20 Address: 0FFD2 <sub>H</sub>

#### ● How to enable/disable/clear interrupts

The interrupt request enable bit (WPCR:WTIE) is used to enable interrupts.

Control item	Interrupt request enable bit (WTIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The watch interrupt request flag (WPCR:WTIF) is used to clear interrupt requests.

Control item	Watch interrupt request flag (WTIF)
To clear an interrupt request	Set the bit to "0".





# ***CHAPTER 13***

---

# ***WATCH COUNTER***

**This chapter describes the functions and operations of the watch counter.**

- 13.1 Overview of Watch Counter
- 13.2 Configuration of Watch Counter
- 13.3 Registers of Watch Counter
- 13.4 Interrupts of Watch Counter
- 13.5 Explanation of Watch Counter Operations and Setup  
Procedure Example
- 13.6 Notes on Using Watch Counter
- 13.7 Sample Programs for Watch Counter

## 13.1 Overview of Watch Counter

The watch counter can generate interrupt requests ranging from min. 125ms to max. 63s intervals.

### ■ Watch Counter

The watch counter performs counting for the number of times specified in the register by using the selected count clock and generates an interrupt request. The count clock can be selected from the four types shown in Table 13.1-1. The count value can be set to any number from 0 to 63. "When "0" is selected, no interrupt is generated.

When the count cycle is set to 1s and the count value is set to "60", an interrupt is generated every one minute.

**Table 13.1-1 Count Clock Types**

Count clock	Count cycle when $F_{CL}$ operates at 32.768kHz
$2^{12}/F_{CL}$	125ms
$2^{13}/F_{CL}$	250 ms
$2^{14}/F_{CL}$	500 ms
$2^{15}/F_{CL}$	1s

$F_{CL}$ : sub clock

**Figure 13.2-1 shows the block diagram of the watch counter.**

### Figure 13.2-1 Block Diagram of Watch Counter



- Counter

This is a 6-bit down-counter that uses the output clock of the watch prescaler as its count clock.

- Watch counter control register (WCSR)

This register controls interrupts and checks the status.

- Watch counter data register (WCDR)

This register sets the interval time and selects the count clock.

■ **Input Clock**

The watch counter uses the output clock of the watch prescaler as its input clock (count clock).

## MB95110B/M Series

### 13.3 Registers of Watch Counter

Figure 13.3-1 shows the registers of the watch counter.

#### ■ Registers of Watch Counter

Figure 13.3-1 Registers Related to Watch Counter

Watch counter data register (WCDR)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FE3 <sub>H</sub>	CS1	CS0	RCTR5	RCTR4	RCTR3	RCTR2	RCTR1	RCTR0	00111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Watch counter control register (WCSR)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0070 <sub>H</sub>	ISEL	WCFLG	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0	00000000 <sub>B</sub>
	R/W	R(RM1)/W	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W:	Readable/writable (Read value is the same as write value)
R(RM1),W :	Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)
R/WX :	Read only (Readable, writing has no effect on operation)

## 13.3.1 Watch Counter Data Register (WCDR)

The watch counter data register (WCDR) is used to select the count clock and set the counter reload value.

### ■ Watch Counter Data Register (WCDR)

Figure 13.3.1-1 Watch Counter Data Register (WCDR)

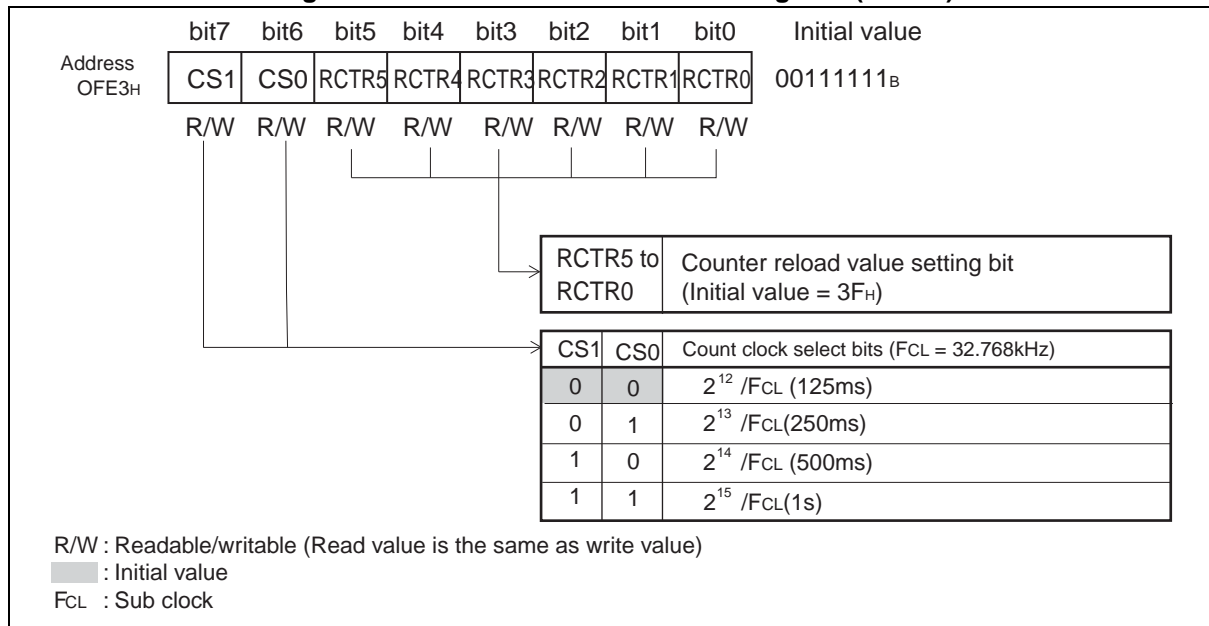


Table 13.3.1-1 Functional Description of Each Bit of Watch Counter Data Register (WCDR)

Bit name		Function
bit7, bit6	CS1, CS0: Count clock select bits	These bits select the clock for the watch counter. $00_B = 2^{12}/F_{CL}$ , $01_B = 2^{13}/F_{CL}$ , $10_B = 2^{14}/F_{CL}$ , $11_B = 2^{15}/F_{CL}$ (F <sub>CL</sub> : sub clock) These bits should be modified when the WCSR:ISEL bit is "0".
bit5 to bit0	RCTR5 to RCTR0: Counter reload value setting bits	These bits set the counter reload value. If the value is modified during counting, the modified value will become effective upon a reload after the counter underflows. When set to "0": No interrupt requests will be generated. If the reload value (RCTR5 to RCTR0) is modified at the same time as an interrupt is generated (WCSR:WCFLG = 1), the correct value will not be reloaded. Therefore, the reload value must be modified before an interrupt is generated, such as when the watch counter is stopped (WCSR:ISEL=0), during the interrupt routine.

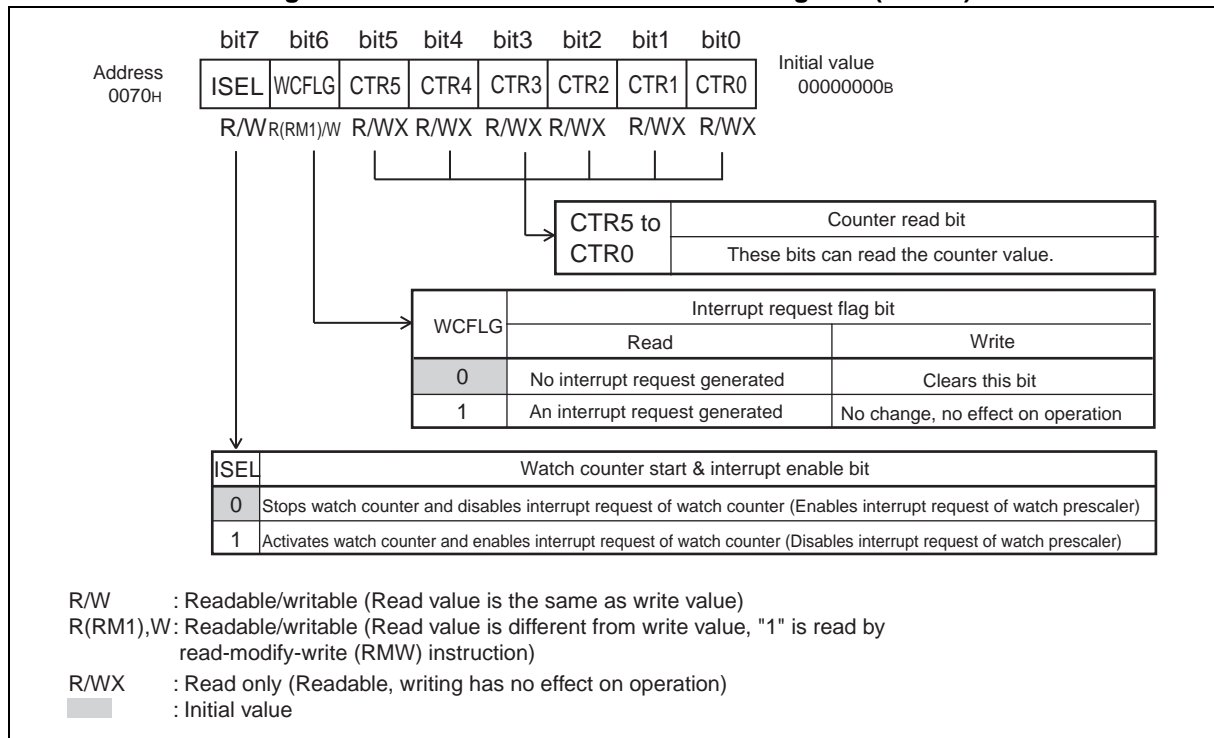
## MB95110B/M Series

### 13.3.2 Watch Counter Control Register (WCSR)

The watch counter control register (WCSR) is used to control the operation and interrupts of the watch counter. It can also read the count value.

#### ■ Watch Counter Control Register (WCSR)

Figure 13.3.2-1 Watch Counter Control Register (WCSR)





**Table 13.3.2-1 Functional Description of Each Bit of Watch Counter Status Register (WCSR)**

Bit name		Function
bit7	ISEL: Watch counter start & interrupt request enable bit	<p>This bit activates the watch counter and selects whether to enable interrupts of the watch counter or those of the watch prescaler.</p> <p>When set to "0": The watch counter is cleared and stopped. Moreover, interrupt requests of the watch counter are disabled, while interrupt requests of the watch prescaler are enabled.</p> <p>When set to "1": The interrupt request output of the watch counter is enabled and the counter starts operation. On the other hand, interrupt requests of the watch prescaler are disabled.</p> <ul style="list-style-type: none"> <li>Always disable interrupts of the watch prescaler before setting this bit to "1" to select interrupts of the watch counter.</li> <li>The watch counter performs counting, using an asynchronous clock from the watch prescaler. For this reason, an error of up to one count clock may occur at the beginning of a count cycle, depending on the timing for setting ISEL bit to "1".</li> </ul>
bit6	WCFLG: Interrupt request flag bit	<p>This bit is set to "1" when the counter underflows.</p> <ul style="list-style-type: none"> <li>When this bit and the ISEL bit are both set to "1", a watch counter interrupt is generated.</li> <li>Writing "0" clears the bit.</li> <li>Writing "1" to this bit has no effects on the operation.</li> <li>"1" is always read in read-modify-write (RMW) instruction.</li> </ul>
bit5 to bit0	CTR5 to CTR0: Counter read bits	<ul style="list-style-type: none"> <li>These bits can read the counter value during counting.</li> </ul> <p>It should be noted that the correct counter value may not be read if a read is attempted while the counter value is being changed. Therefore, read the counter value twice to check if the same value is read on both occasions before using it.</p> <ul style="list-style-type: none"> <li>Writing has no effect on the operation.</li> </ul>

## 13.4 Interrupts of Watch Counter

The watch counter outputs interrupt requests when the counter underflows (counter value = 000001<sub>B</sub>).

### ■ Interrupts of Watch Counter

When the counter of the watch counter underflows, the interrupt request flag bit (WCFLG) of the watch counter control register (WCSR) is set to "1". If the interrupt request enable bit (ISEL) of the watch counter is set to "1", an interrupt request of the watch counter is outputted to the interrupt controller.

Table 13.4-1 shows the interrupt control bits and interrupt sources of the watch timer.

**Table 13.4-1 Interrupt Control Bits and Interrupt Sources of Watch Timer**

Item	Description
Interrupt request flag bit	WCFLG bit of the WCSR register
Interrupt request enable bit	ISEL bit of the WCSR register
Interrupt source	Counter underflow

### ■ Register and Vector Table Related to Interrupts of Watch Counter

**Table 13.4-2 Register and Vector Table Related to Interrupts of Watch Counter**

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Watch counter*	IRQ20	ILR5	L20	FFD2 <sub>H</sub>	FFD3 <sub>H</sub>

\*: The watch counter shares the same interrupt request number and vector table as the watch prescaler.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

## 13.5 Explanation of Watch Counter Operations and Setup Procedure Example

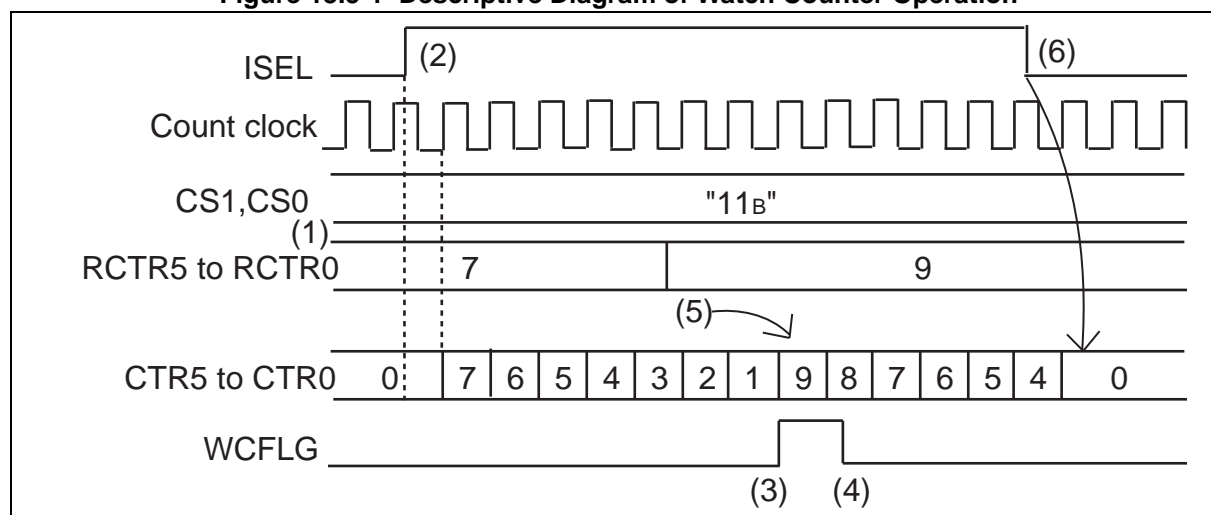
The watch counter counts down for the number of times specified in the count value by RCTR5 to RCTR0 bits, using the count clock selected by CS1 and CS0 bits, when the ISEL bit is set to "1". Once the counter underflows, WCFLG bit of the WCSR register is set to "1", generating an interrupt.

### ■ Setup Procedure of Watch Counter

The setup procedure of the watch counter is described below.

- (1) Select the count clock (CS1 and CS0 bits) and set the counter reload value (RCTR5 to RCTR0 bits).
- (2) Set the ISEL bit of the WCSR register to "1" to start a down count and enable interrupts. Also disable interrupts of the watch prescaler.  
The watch counter performs counting by using a divided clock (asynchronous) from the watch prescaler. An error of up to one count clock may occur at the beginning of a count cycle, depending on the timing for setting the ISEL bit to "1".
- (3) When the counter underflows, the WCFLG bit of the WCSR register is set to "1", generating an interrupt.
- (4) Write "0" to the WCFLG bit to clear it.
- (5) If RCTR5 to RCTR0 bits are modified during counting, the reload value will be updated during a reload after the counter is set to "1".
- (6) When writing "0" to the ISEL bit, the counter becomes "0" and stops operation.

Figure 13.5-1 Descriptive Diagram of Watch Counter Operation



Note:

When the operation is reactivated by WCSR:ISEL=0 after counter stop, please reactivate after confirming reading WCSR:CTR[5:0] twice, and clearing to CTR[5:0]=000000<sub>B</sub>.

**■ Operation in Sub Clock Stop Mode**

When the device enters the sub clock stop mode, the watch counter stops the count operation and the watch prescaler is also cleared. Therefore, the watch counter cannot count the correct value after the sub clock stop mode is cancelled. After the sub clock stop mode is cancelled, the ISEL bit must always be set to "0" to clear the counter before use. In any standby mode other than the sub clock stop mode, the watch counter continues to operate.

**■ Operation at the Main Clock Stop Mode**

The interrupt is not generated though the clock counter continues the count operation when entering the main clock stop mode. Moreover, the clock counter stops, too, when sub clock oscillation stop bit (SYCC: SUBS) of the system clock control register is set to "1".

**■ Setup Procedure Example**

The watch counter is set up in the following procedure:

**● Initial setting**

- 1) Set the interrupt level. (ILR5)
- 2) Select the count clock. (WCDR:CS1, CS0)
- 3) Set the counter reload value. (WCDR:RCTR5 to RCTR0)
- 4) Activate the watch counter and enable interrupts.(WCSR:ISEL=1)

**● Interrupt processing**

- 1) Clear the interrupt request flag.(WCSR:WCFLG=0)
- 2) Arbitrary processing

## **13.6 Notes on Using Watch Counter**

---

**Shown below are the precautions that must be followed when using the watch counter.**

---

- If the watch prescaler is cleared during the operation of the watch counter, the watch counter may not be able to perform normal operation. When clearing the watch prescaler, set the ISEL bit of the WCSR register to "0" to stop the watch counter in advance.
- When the operation is reactivated by WCSR:ISEL=0 after counter stop, please reactivate after confirming reading WCSR:CTR[5:0] twice, and clearing to CTR[5:0]=000000<sub>B</sub>.

## 13.7 Sample Programs for Watch Counter

We provide sample programs that can be used to operate the watch counter.

### ■ Sample Programs for Watch Counter

For information about sample programs for the watch counter, refer to "■ Sample Programs" in Preface.

### ■ Setting Methods not Covered by Sample Programs

#### ● How to enable/stop the watch counter

Use the interrupt request enable bit (WCSR:ISEL).

Control item	Watch timer initialization bit (ISEL)
When enabling watch counter	Set the bit to "1".
When stopping watch counter	Set the bit to "0".

#### ● How to select the count clock

The count clock select bits (WCDR:CS1/CS0) are used to select the clock.

#### ● Interrupt-related register

The interrupt level is set in the interrupt level register shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
Watch counter	Interrupt level register (ILR5) Address: 0007E <sub>H</sub>	#20 Address: 0FFD2 <sub>H</sub>

#### ● How to enable/disable/clear interrupts

The interrupt request enable bit (WCSR:ISEL) is used to enable interrupts.

Control item	Interrupt request enable bit (ISEL)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

The interrupt request flag (WCSR:WCFLG) is used to clear interrupt requests.

Control item	Interrupt request flag (WCFLG)
To clear an interrupt request	Set the bit to "0".



# ***CHAPTER 14***

---

# ***WILD REGISTER***

**This chapter describes the functions and operations of the wild register.**

- 14.1 Overview of Wild Register
- 14.2 Configuration of Wild Register
- 14.3 Registers of Wild Register
- 14.4 Operating Description of Wild Register
- 14.5 Typical Hardware Connection Example



## **14.1 Overview of Wild Register**

---

**The wild register can be used to patch bugs in the program by using the addresses set in the built-in register and amendment data.**

**The following section describes the wild register function.**

---

### **■ Wild Register Function**

The wild register consists of 3 data setup registers, 3 upper-address setup registers, 3 lower-address setup registers, a 1-byte address compare enable register and a 1-byte data test setup register. When certain addresses and modified data are specified in these registers, the ROM data can be replaced with the modified data specified in the registers. Data of up to three different addresses can be modified.

The wild register function can be used to debug the program after creating the mask and patch bugs in the program.

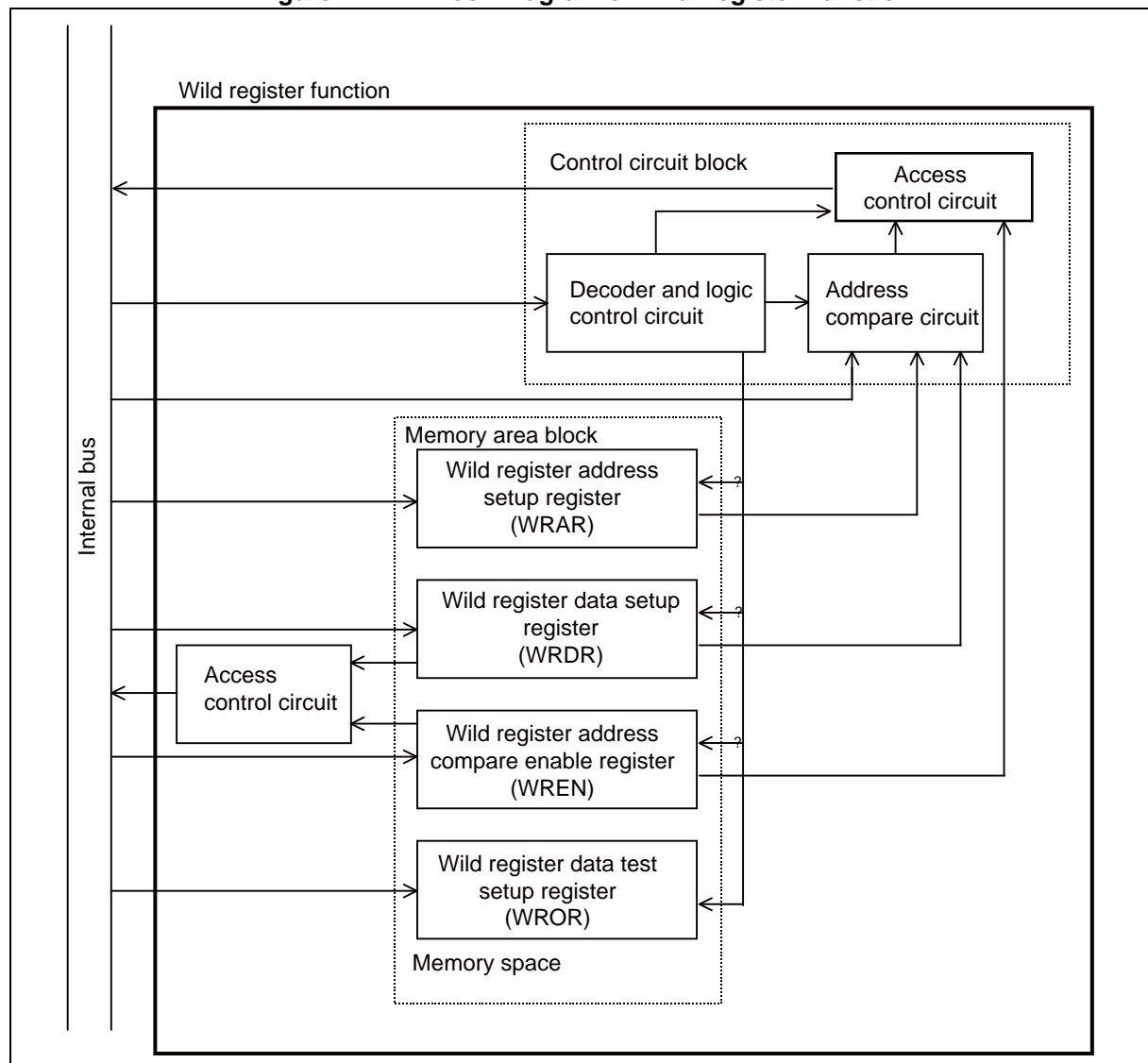
## 14.2 Configuration of Wild Register

The block diagram of the wild register is shown below. The wild register consists of the following blocks:

- **Memory area block**
  - Wild register data setup register (WRDR0 to WRDR2)
  - Wild register address setup register (WRAR0 to WRAR2)
  - Wild register address compare enable register (WREN)
  - Wild register data test setup register (WROR)
- **Control circuit block**

### ■ Block Diagram of Wild Register Function

Figure 14.2-1 Block Diagram of Wild Register Function



● Memory area block

The memory area block consists of the wild register data setup registers (WRDR), wild register address setup registers (WRAR), wild register address compare enable register (WREN) and wild register data test setup register (WROR). The wild register function is used to specify the addresses and data that need to be replaced. The wild register address compare enable register (WREN) enables the wild register function for each wild register data setup register (WRDR). Moreover, the wild register data test setup register (WROR) enables the normal read function for each wild register data setup register (WRDR).

● Control circuit block

This circuit compares the actual address data with addresses set in the wild register address setup registers (WRAR), and if the values match, outputs the data from the wild register data setup register (WRDR) to the data bus. The control circuit block uses the wild register address compare enable register (WREN) to control the operation.

## MB95110B/M Series

### 14.3 Registers of Wild Register

The registers of the wild register include the wild register data setup registers (WRDR), wild register address setup registers (WRAR), wild register address compare enable register (WREN) and wild register data test setup register (WROR).

#### ■ Registers Related to Wild Register

Figure 14.3-1 Registers Related to Wild Register

Wild register data setup registers (WRDR0 to WRDR2)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WRDR0	0F82 <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000 <sub>B</sub>
WRDR1	0F85 <sub>H</sub>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRDR2	0F88 <sub>H</sub>									
Wild register address setup registers (WRAR0 to WRAR2)										
	Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
WRAR0	0F80 <sub>H</sub> , 0F81 <sub>H</sub>	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000 <sub>B</sub>
WRAR1	0F83 <sub>H</sub> , 0F84 <sub>H</sub>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRAR2	0F86 <sub>H</sub> , 0F87 <sub>H</sub>									
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
		RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000 <sub>B</sub>
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Wild register address compare enable register (WREN)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WREN	0076 <sub>H</sub>	—	—	Reserved	Reserved	Reserved	EN2	EN1	EN0	00000000 <sub>B</sub>
		R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	
Wild register data test setup register (WROR)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
WROR	0077 <sub>H</sub>	—	—	Reserved	Reserved	Reserved	DDR2	DDR1	DDR0	00000000 <sub>B</sub>
		R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	
R/W : Readable/writable (Read value is the same as write value) R0/W0 : Reserved bit (Write value is "0", read value is "0") R0/WX : Undefined bit (Read value is "0", writing has no effect on operation) — : Undefined										

■ **Wild Register Number**

Each wild register address setup register (WRAR) and wild register data setup register (WRDR) has its corresponding wild register number.

**Table 14.3-1 Wild Register Numbers Corresponding to Wild Register Address Setup Registers and Wild Register Data Setup Registers**

Wild register number	Wild registers address setup register (WRAR)	Wild registers data setup register (WRDR)
0	WRAR0	WRDR0
1	WRAR1	WRDR1
2	WRAR2	WRDR2

## MB95110B/M Series

### 14.3.1 Wild Register Data Setup Registers (WRDR0 to WRDR2)

The wild register data setup registers (WRDR0 to WRDR2) use the wild register function to specify the data to be amended.

#### ■ Wild Register Data Setup Registers (WRDR0 to WRDR2)

Figure 14.3-2 Wild Register Data Setup Registers (WRDR0 to WRDR2)

WRDR0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F82 <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRDR1									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F85 <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRDR2									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F88 <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

Table 14.3-2 Functional Description of Each Bit of Wild Register Data Setup Register (WRDR0 to WRDR2)

Bit name		Function
bit7 to bit0	RD7 to RD0: Wild registers data setup bits	<p>These bits specify the data to be amended by the wild register function.</p> <ul style="list-style-type: none"> <li>These bits are used to set the amendment data at the address assigned by the wild register address setup register (WRAR). Data is enabled at the address corresponding to each wild register number.</li> <li>Read access of these bits is enabled only when the corresponding data test setting bit in the wild register data test setup register (WROR) is set to "1".</li> </ul>

## 14.3.2 Wild Register Address Setup Registers (WRAR0 to WRAR2)

The wild register address setup registers (WRAR0 to WRAR2) set the address to be amended by the wild register function.

### ■ Wild Register Address Setup Registers (WRAR0 to WRAR2)

Figure 14.3-3 Wild Register Address Setup Registers (WRAR0 to WRAR2)

WRAR0									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0F80 <sub>H</sub>	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F81 <sub>H</sub>	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRAR1									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0F83 <sub>H</sub>	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F84 <sub>H</sub>	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
WRAR2									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0F86 <sub>H</sub>	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0F87 <sub>H</sub>	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									

Table 14.3-3 Functional Description of Each Bit of Wild Register Address Setup Register (WRAR0 to WRAR2)

Bit name		Function
bit15 to bit0	RA15 to RA0: Wild Registers address setting bits	These bits set the address to be amended by the wild register function. These bits are used to specify the address to be allocated. The address is specified in accordance with its corresponding wild register number.

## MB95110B/M Series

### 14.3.3 Wild Register Address Compare Enable Register (WREN)

The wild register address compare enable register (WREN) enables/disables the operation of the wild register in accordance with each wild register number.

#### ■ Wild Register Address Compare Enable Register (WREN)

Figure 14.3-4 Wild Register Address Compare Enable Register (WREN)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0076 <sub>H</sub>	—	—	Reserved	Reserved	Reserved	EN2	EN1	EN0	00000000 <sub>B</sub>
	R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)  
 R0/W0 : Reserved bit (Write value is "0", read value is "0")  
 R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)  
 — : Undefined

Table 14.3-4 Functional Description of Wild Register Address Compare Enable Register (WREN)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. <ul style="list-style-type: none"> <li>The read value is "0".</li> <li>Writing has no effect on the operation.</li> </ul>
bit5 to bit3	Reserved bit	These bits are reserved. <ul style="list-style-type: none"> <li>The read value is "0".</li> <li>Always set "0".</li> </ul>
bit2 to bit0	EN2, EN1, EN0: Wild register address compare enable bits	These bits enable/disable the operation of the wild register. <ul style="list-style-type: none"> <li>EN0 corresponds to wild register number 0.</li> <li>EN1 corresponds to wild register number 1.</li> <li>EN2 corresponds to wild register number 2.</li> </ul> When set to "0": disable the operation of the wild register function. When set to "1": enable the operation of the wild register function.



## 14.3.4 Wild Register Data Test Setup Register (WROR)

The wild register data test setup register (WROR) enables/disables reading from the corresponding wild register data setup register (WRDR0 to WRDR2).

### ■ Wild Register Data Test Setup Register (WROR)

Figure 14.3-5 Wild Register Data Test Setup Register (WROR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0077 <sub>H</sub>	—	—	Reserved	Reserved	Reserved	DDR2	DDR1	DDR0	00000000 <sub>B</sub>
	R0/WX	R0/WX	R0/W0	R0/W0	R0/W0	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)  
 R0/W0 : Reserved bit (Write value is "0", read value is "0")  
 R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)  
 — : Undefined

Table 14.3-5 Functional Description of Wild Register Data Test Setup Register (WROR)

Bit name		Function
bit7, bit6	Undefined bits	These bits are undefined. <ul style="list-style-type: none"> <li>The read value is "0".</li> <li>Writing has no effect on the operation.</li> </ul>
bit5 to bit3	Reserved bits	These bits are reserved. <ul style="list-style-type: none"> <li>The read value is "0".</li> <li>Always set "0".</li> </ul>
bit2 to bit0	DDR2, DDR1, DDR0: Wild registers data test setup bits	These bits enable/disable the normal reading from the corresponding data setup register of the wild register. <ul style="list-style-type: none"> <li>DDR0 enables/disables reading from the wild register data setup register (WRDR0).</li> <li>DDR1 enables/disables reading from the wild register data setup register (WRDR1).</li> <li>DDR2 enables/disables reading from the wild register data setup register (WRDR2).</li> </ul> When set to "0": disable reading. When set to "1": enable reading.

## 14.4 Operating Description of Wild Register

This section describes the setup procedure for the wild register.

### ■ Setup Procedure for Wild Register

Prepare a special program that can read the value to be set in the wild register from external memory (e.g. E<sup>2</sup>PROM or FRAM) in the user program before executing the program. The setup method for the wild register is shown below.

It should be noted that this section does not explain how to communicate between the external memory and the device.

- Write the address of the built-in ROM code that will be modified to the wild register address setup register (WRAR0 to WRAR2).
- Write a new code into the corresponding wild register data setup register (WRDR0 to WRDR2).
- Write the corresponding bits to the wild register address compare enable register (WREN) to enable the wild register function.

Table 14.4-1 shows the register setup procedure for the wild register.

**Table 14.4-1 Register Setup Procedure for Wild Register**

Operating step	Operation	Example operation
1	Read replacement data from outside through its specific communication method.	The built-in ROM code to be modified is in the address "F011 <sub>H</sub> " and the data to be modified is "B5 <sub>H</sub> ". Three built-in ROM codes can be modified.
2	Write the replacement address into the wild register address setup register (WRAR0 to WRAR2).	Set Wild register address setup registers (WRAR0 = F011 <sub>H</sub> , WRAR1 = ..., WRAR2 = ...).
3	Write a new ROM code (replacement for the built-in ROM code) to the wild register data setup register (WRDR0 to WRDR2).	Set Wild register data setup registers (WRDR0 = B5 <sub>H</sub> , WRDR1 = ..., WRDR2 = ...).
4	Enable the corresponding bits in the wild register address compare enable register (WREN).	Setting bit0 of the address compare enable register (WREN) to "1" enables the wild register function for the wild register number 0. If the address matches the value set in the address setup register (WRAR), the value of the data setup register (WRDR) will replace the built-in ROM code. When replacing more than one built-in ROM code, enable the corresponding bits of the address compare enable register (WREN).

### ■ Wild Register Applicable Addresses

The wild register is applicable to all addresses in the address space except "0078<sub>H</sub>".

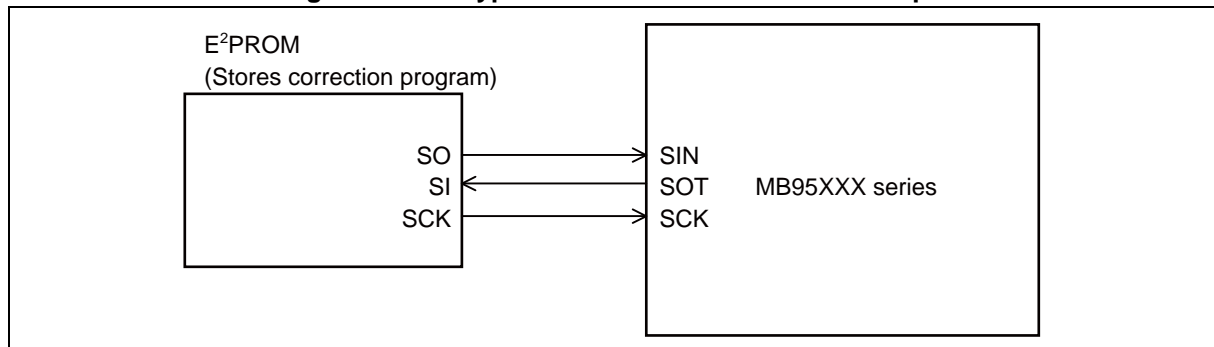
As address "0078<sub>H</sub>" is used as a mirror address for the register bank pointer and direct bank pointer, this address cannot be patched.

## 14.5 Typical Hardware Connection Example

Shown below is a typical hardware connection example applied when using the wild register function.

### ■ Hardware Connection Example

**Figure 14.5-1 Typical Hardware Connection Example**



# CHAPTER 15

---

## 8/16-BIT COMPOUND TIMER

**This chapter describes the functions and operations of the 8/16-bit compound timer.**

- 15.1 Overview of 8/16-bit Compound Timer
- 15.2 Configuration of 8/16-bit Compound Timer
- 15.3 Channels of 8/16-bit Compound Timer
- 15.4 Pins of 8/16-bit Compound Timer
- 15.5 Registers of 8/16-bit Compound Timer
- 15.6 Interrupts of 8/16-bit Compound Timer
- 15.7 Operating Description of Interval Timer Function (One-shot Mode)
- 15.8 Operating Description of Interval Timer Function (Continuous Mode)
- 15.9 Operating Description of Interval Timer Function (Free-run Mode)
- 15.10 Operating Description of PWM Timer Function (Fixed-cycle mode)
- 15.11 Operating Description of PWM Timer Function (Variable-cycle Mode)
- 15.12 Operating Description of PWC Timer Function
- 15.13 Operating Description of Input Capture Function
- 15.14 Operating Description of Noise Filter
- 15.15 States in Each Mode during Operation
- 15.16 Notes on Using 8/16-bit Compound Timer

## 15.1 Overview of 8/16-bit Compound Timer

---

The 8/16-bit compound timer consists of two 8-bit counters and can be used as two 8-bit timers, or one 16-bit timer if they are connected in cascade.

The 8/16-bit compound timer has the following functions:

- Interval timer function
  - PWM timer function
  - PWC timer function (pulse width measurement)
  - Input capture function
- 

### ■ Interval Timer Function (One-shot Mode)

When the interval timer function (one-shot mode) is selected, the counter starts counting from "00<sub>H</sub>" as the timer is started. When the counter value matches the register setting value, the timer output is inverted, the interrupt request occurs, and the count operation is stopped.

### ■ Interval Timer Function (Continuous Mode)

When the interval timer function (continuous mode) is selected, the counter starts counting from "00<sub>H</sub>" as the timer is started. When the counter value matches the register setting value, the timer output is inverted, the interrupt request occurs, and the count operation is continued from "00<sub>H</sub>" again. The timer output a square wave as a result of this repeated operation.

### ■ Interval Timer Function (Free-run Mode)

When the interval timer function (free-run mode) is selected, the counter starts counting from "00<sub>H</sub>". When the counter value matches the register setting value, the timer output is inverted and the interrupt request occurs. When the counter continues to count until reaching "FF<sub>H</sub>", it restarts counting from "00<sub>H</sub>" to continue the counting operation. The timer outputs a square wave as a result of this repeated operation.

### ■ PWM Timer Function (Fixed-cycle Mode)

When the PWM timer function (fixed-cycle mode) is selected, a PWM signal with a variable "H" pulse width is generated in fixed cycles. The cycle is fixed to "FF<sub>H</sub>" during 8-bit operation or "FFFF<sub>H</sub>" during 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by setting a register.

### ■ PWM Timer Function (Variable-cycle Mode)

When the PWM timer function (variable-cycle mode) is selected, two 8-bit counters are used to generate an 8-bit PWM signal in any cycles and duty depending on the cycle and "L" pulse width specified by registers. In this operation mode, the compound timer cannot serve as a 16-bit counter, as two 8-bit counters are used.

#### ■ PWC Timer Function

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured.

In this operation mode, the counter starts counting from "00<sub>H</sub>" upon detection of a count start edge of an external input signal and transfers the count value to a register to generate an interrupt upon detection of a count end edge.

#### ■ Input Capture Function

When the input capture function is selected, the counter value is stored in a register upon detection of an edge for an external input signal.

This function is available in either free-run mode or clear mode for count operation.

In the clear mode, the counter starts counting from "00<sub>H</sub>" and transfers its value to a register to generate an interrupt upon detection of an edge. In this case, the counter continues to count from "00<sub>H</sub>".

In the free-run mode, the counter transfers its value to a register to generate an interrupt upon detection of an edge. In this case, however, the counter continues to count without being cleared.

## **15.2 Configuration of 8/16-bit Compound Timer**

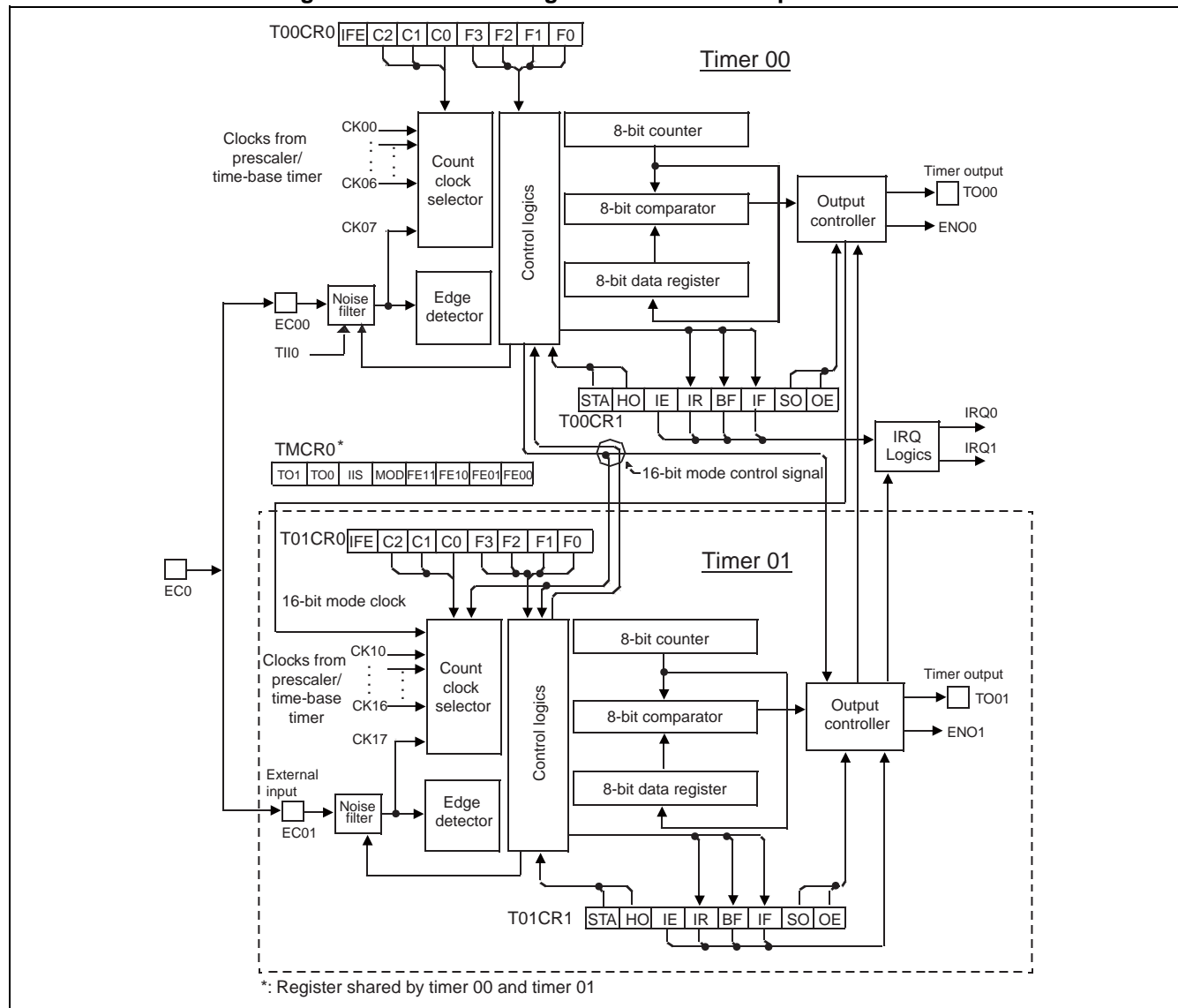
---

The 8/16-bit compound timer consists of the following blocks:

- 8-bit counter  $\times$  2 channels
  - 8-bit comparator (including a temporary latch)  $\times$  2 channels
  - 8/16-bit compound timer 00/01 data register  $\times$  2 channels (T00DR/T01DR)
  - 8/16-bit compound timer 00/01 control status register 0  $\times$  2 channels (T00CR0/T01CR0)
  - 8/16-bit compound timer 00/01 control status register 1  $\times$  2 channels (T00CR1/T01CR1)
  - 8/16-bit compound timer 00/01 timer mode control register (TMCR0)
  - Output controller  $\times$  2 channels
  - Control logic  $\times$  2 channels
  - Count clock selector  $\times$  2 channels
  - Edge detector  $\times$  2 channels
  - Noise filter  $\times$  2 channels
-

■ Block Diagram of 8/16-bit Compound Timer

Figure 15.2-1 Block Diagram of 8/16-bit Compound Timer



● 8-bit counter

This counter serves as the basis for various timer operations. It can be used either as two 8-bit counters or as a 16-bit counter.

● 8-bit comparator

The comparator compares the values in the 8/16-bit compound timer 00/01 data register and counter. It incorporates a latch to temporarily store the 8/16-bit compound timer 00/01 data register value.

● 8/16-bit compound timer 00/01 data register

The 8/16-bit compound timer 00/01 data register is used to write the maximum value counted during interval timer or PWM timer operation and to read the count value during PWC timer or input capture operation.



- 8/16-bit compound timer 00/01 control status registers 0 (T00CR0/T01CR0)

These registers are used to select the timer operation mode, select the count clock, and to enable or disable IF flag interrupts.

- 8/16-bit compound timer 00/01 control status registers 1 (T00CR1/T01CR1)

These registers are used to control interrupt flags, timer output, and timer operation.

- 8/16-bit compound timer 00/01 timer mode control register (TMCR0)

This register is used to select the noise filter function, 8-bit or 16-bit operation mode, and signal input to timer 00 and to indicate the timer output value.

- Output controller

The output controller controls timer output. The timer output is supplied to the external pin when the pin output has been enabled.

- Control logic

The control logic controls timer operation.

- Count clock selector

The selector selects the counter operation clock signal from among prescaler outputs (machine clock divided signal and time-base timer output).

- Edge detector

The edge detector selects the edge of an external input signal to be used as an event for PWC timer operation or input capture operation.

- Noise filter

This filter serves as a noise filter for external input signals. "H" pulse noise, "L" pulse noise, or "H"/"L"-pulse noise elimination can be selected as the filter function.

- TII0 internal pin (internally connected to the LIN-UART, available only in channel 0)

The TII0 pin serves as the signal input pin for timer 00; it is connected to the LIN-UART inside the chip. For information about how to use the pin, refer to "CHAPTER 22 LIN-UART". Note that the TII0 pin in channel 1 is internally fixed to "0".

## ■ Input Clock

The 8/16-bit compound timer uses the output clock from the prescaler as its input clock (count clock).

## 15.3 Channels of 8/16-bit Compound Timer

This section describes the channels of 8/16-bit compound timer.

### ■ Channels of 8/16-bit Compound Timer

MB95110B/M series contains two channels of 8/16-bit compound timer.

In one channel, there are two 8-bit counters. Each counter can be used as two 8-bit timers or one 16-bit timer. The following table lists the external pins and registers corresponding to each channel.

**Table 15.3-1 8/16-bit Compound Timer Channels and Corresponding External Pins**

Channel	Pin name	Pin function
0	TO00	Timer 00 output
	TO01	Timer 01 output
	EC0	Timer 00 input and timer 01 input
1	TO10	Timer 10 output
	TO11	Timer 11 output
	EC1	Timer 10 input and timer 11 input

**Table 15.3-2 8/16-bit Compound Timer Channels and Corresponding Registers**

Channel	Register name	Registers
0	T00CR0	Timer 00 control status register 0
	T01CR0	Timer 01 control status register 0
	T00CR1	Timer 00 control status register 1
	T01CR1	Timer 01 control status register 1
	T00DR	Timer 00 data register
	T01DR	Timer 01 data register
	TMCR0	Timer 00/01 timer mode control register
1	T10CR0	Timer 10 control status register 0
	T11CR0	Timer 11 control status register 0
	T10CR1	Timer 10 control status register 1
	T11CR1	Timer 11 control status register 1
	T10DR	Timer 10 data register
	T11DR	Timer 11 data register
	TMCR1	Timer 10/11 timer mode control register

The following sections describe only the 8/16-bit compound timer in channel 0.

The other channels are the same as channel 0. The 2-digit number in the pin names and register names corresponds to channel and timer. The upper number corresponds to channel and the lower number corresponds to timer.

## 15.4 Pins of 8/16-bit Compound Timer

---

**This section describes the pins related to the 8/16-bit compound timer.**

---

### ■ Pins Related to 8/16-bit Compound Timer

The external pins related to the 8/16-bit compound timer are TO00, TO01, EC0, and EC1. TH0 is for internal chip connection.

#### ● TO00 pins

TO00:

This pin serves as the timer output pin for timer 00 during 8-bit operation or for timers 00 and 01 during 16-bit operation. When the output is enabled (T00CR1:OE = 1) in interval timer, PWM timer, or PWC timer function, the pin is set for output automatically regardless of the port direction register (DDR2:bit2) to serve as the timer output TO00 pin.

The output remains indeterminate when the input capture function has been selected enabling output.

#### ● TO01 pins

TO01:

This pin serves as the timer output pin for timer 01 during 8-bit operation. When the output is enabled (T00CR1:OE = 1) in interval timer, PWM timer (fixed cycle mode), or PWC timer function, the pin is set for output automatically regardless of the port direction register (DDR2:bit3) to serve as the timer output TO01 pin.

The output remains indeterminate during 16-bit operation when the PWM timer function (variable-cycle mode) or input capture function has been selected enabling output.

#### ● EC0 pins

The EC0 pin is connected to the EC00 and EC01 internal pins.

EC00 internal pin:

This pin serves as the external count clock input pin for timer 00 when the interval timer or PWM timer function has been selected, or as the signal input pin for timer 00 when the PWC timer or input capture function has been selected. The pin cannot be set as the external count clock input pin when the PWC timer or input capture function has been selected.

To use this input feature, set the port direction register (DDR2:bit4) to set the pin as an input port.

EC01 internal pin:

This pin serves as the external count clock input pin for timer 01 when the interval timer or PWM timer function has been selected or the signal input pin for timer 01 when the PWC timer or input capture function has been selected. The pin cannot be set as the external count clock input pin when the PWC timer or input capture function has been selected.

This input is not used during 16-bit operation. The input can be used as well when the PWM timer function has been selected (variable-cycle mode).

To use this input feature, set the port direction register (DDR2:bit4) to "0" to set the pin as an input port.

## ■ Block Diagram of Pins Related to 8/16-bit Compound Timer

Figure 15.4-1 Block Diagram of Pin Block Diagram of Pins Related to 8/16-bit Compound Timer(TO00, TO01, EC0)

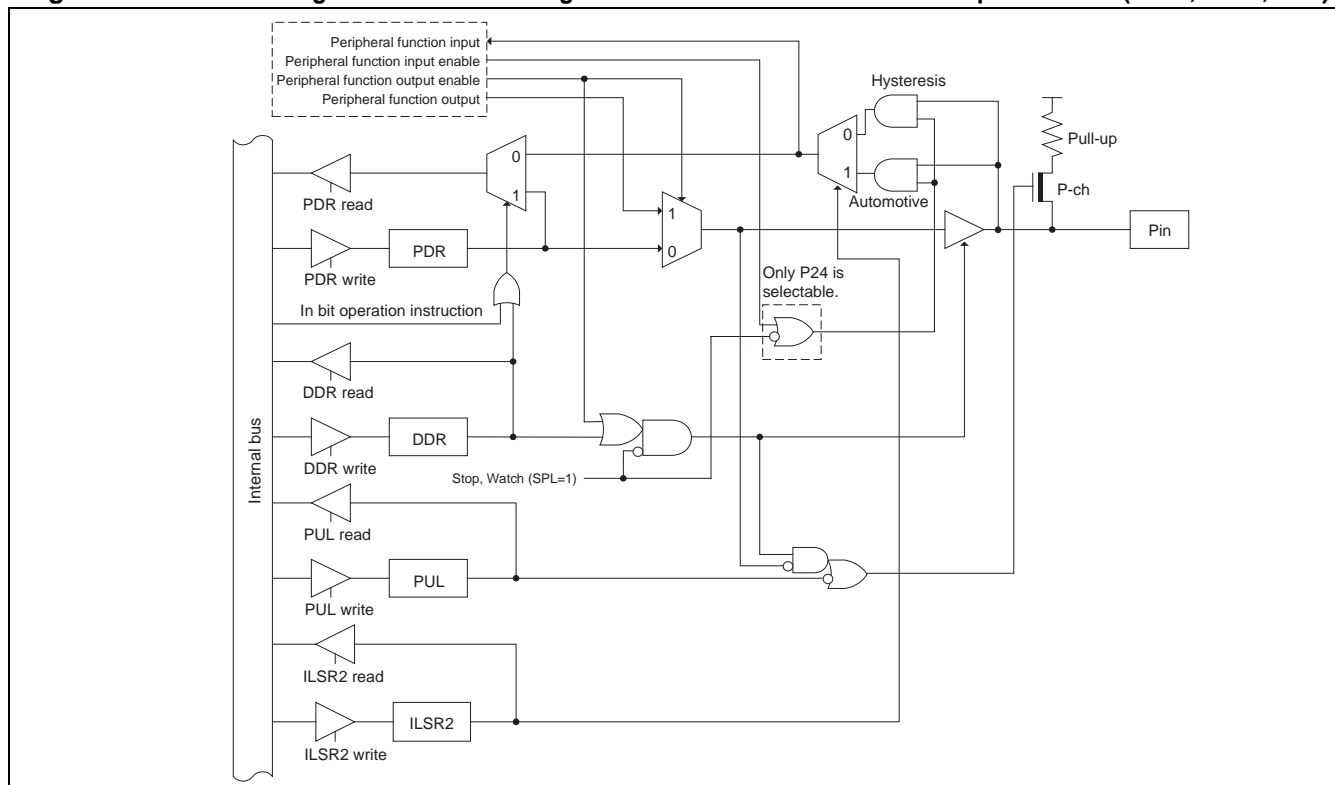
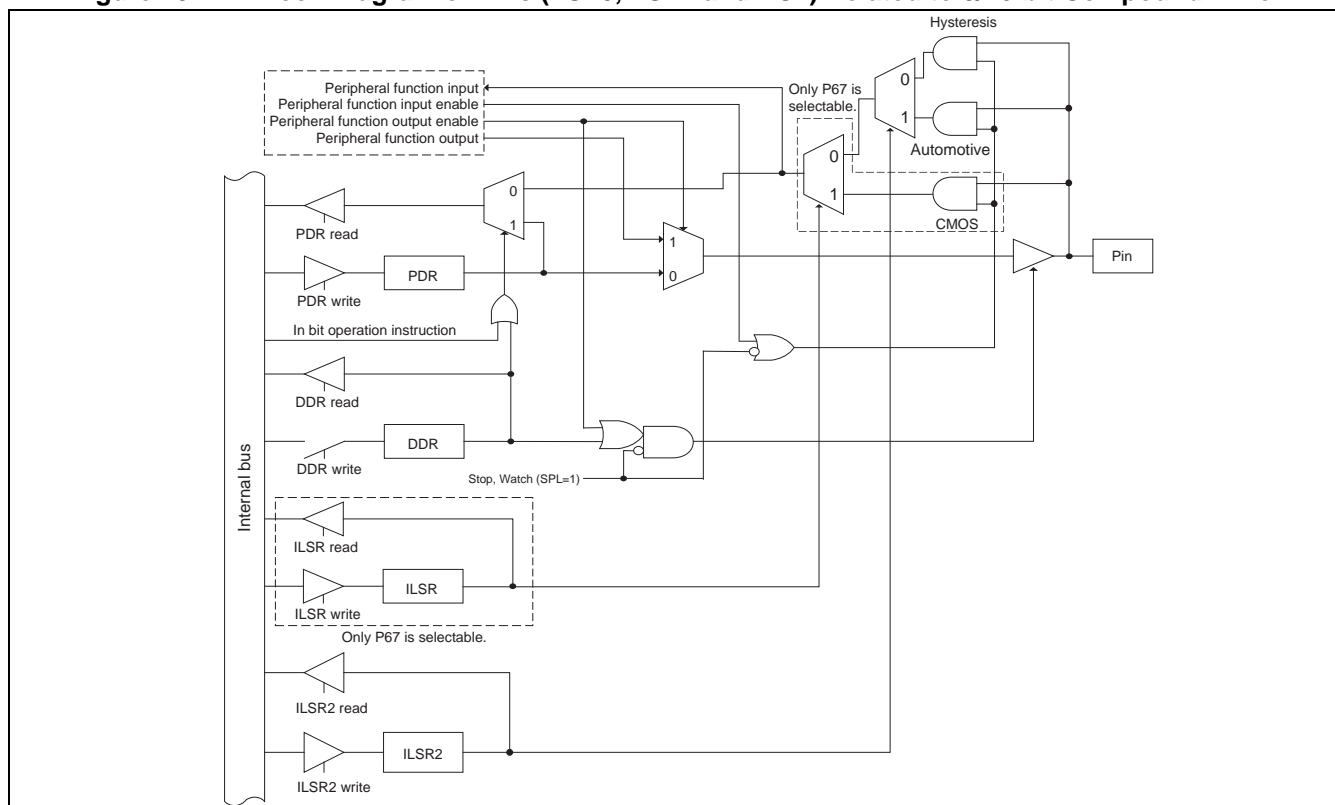


Figure 15.4-2 Block Diagram of Pins (TO10, TO11 and EC1) Related to 8/16-bit Compound Timer



## 15.5 Registers of 8/16-bit Compound Timer

This section describes the registers related to the 8/16-bit compound timer.

### ■ Registers Related to 8/16-bit Compound Timer

Figure 15.5-1 Registers Related to 8/16-bit Compound Timer

8/16-bit compound timer 00/01 control status register 0 (T00CR0/T01CR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01CR0 0F92 <sub>H</sub>	IFE	C2	C1	C0	F3	F2	F1	F0	00000000 <sub>B</sub>
T00CR0 0F93 <sub>H</sub>	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	

8/16-bit compound timer 00/01 control status register 1 (T00CR1/T01CR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01CR 10036 <sub>H</sub>	STA	HO	IE	IR	BF	IF	SO	OE	00000000 <sub>B</sub>
T00CR 10037 <sub>H</sub>	R/W	R/W	R/W	R(RM1),W	R/WX	R(RM1),W	R/W	R/W	

8/16-bit compound timer 00/01 data register (T00DR/T01DR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
T01DR 0F94 <sub>H</sub>	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000 <sub>B</sub>
T00DR 0F95 <sub>H</sub>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/16-bit compound timer 00/01 timer mode control register (TMCR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit0	bit0	Initial value
TMCR0 0F96 <sub>H</sub>	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00	00000000 <sub>B</sub>
	R/WX	R/WX	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)

R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

R/WX : Read only (Readable, writing has no effect on operation)

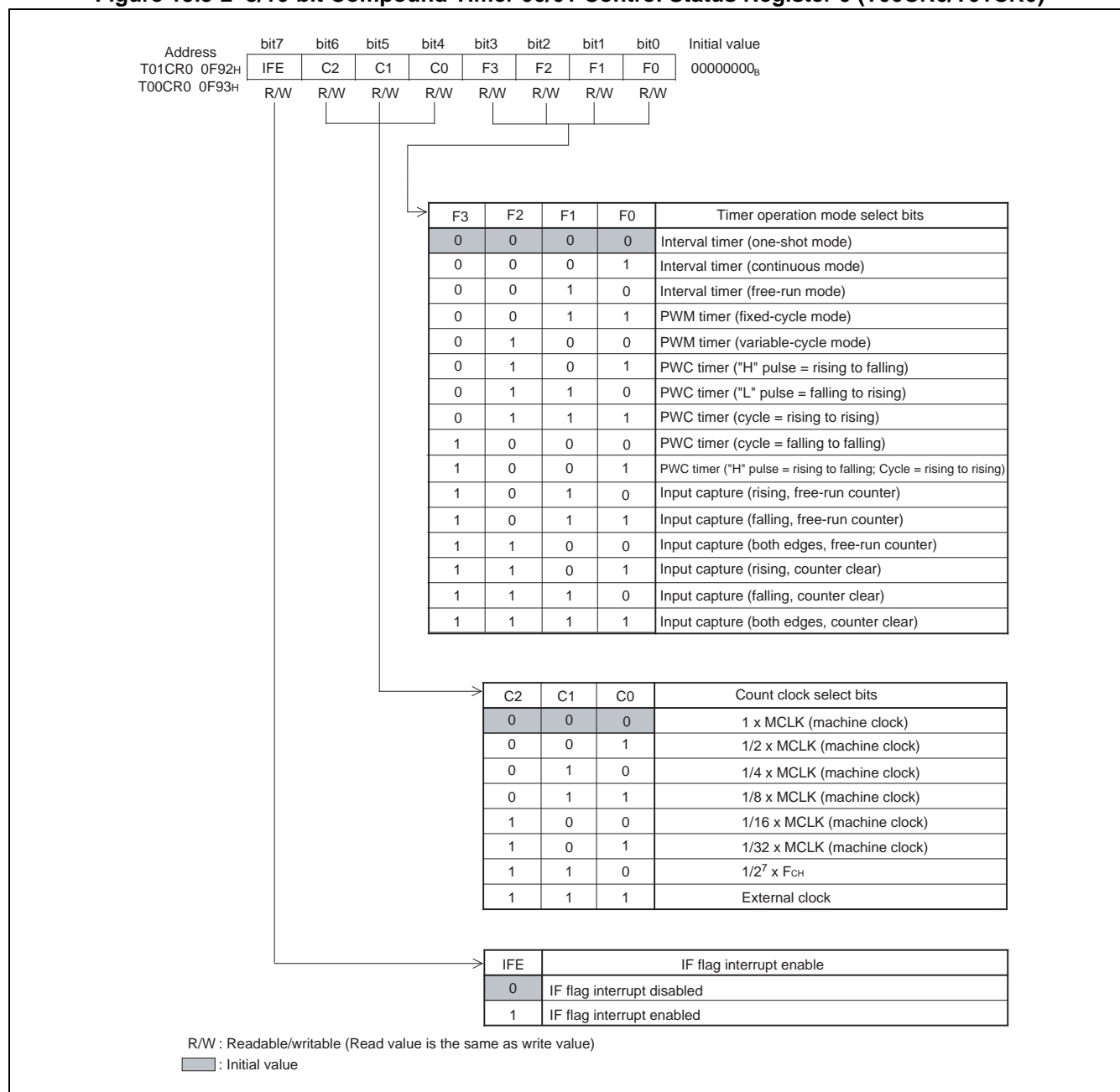
R,W : Readable, writable (Read value is different from write value)

## 15.5.1 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)

The 8/16-bit compound timer 00/01 control status register 0 (T00CR0/T01CR0) selects the timer operation mode, selects the count clock, and enables or disables IF flag interrupts. The T00CR0 and T01CR0 registers correspond to timers 00 and 01, respectively.

### ■ 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)

Figure 15.5-2 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0)



**Table 15.5-1 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) (1 / 2)**

Bit name		Function																																				
bit7	IFE: IF flag interrupt enable	This bit enables or disables IF flag interrupts. <b>Setting this bit to "0"</b> : disables IF flag interrupts. <b>Setting this bit to "1"</b> : an IF flag interrupt request is outputted when both the IE bit (T00CR1/T01CR1:IE) and the IF flag (T00CR1/T01CR1:IF) are set to "1".																																				
bit6 to bit4	C2, C1, C0: Count clock select bits	<p>These bits select the count clock.</p> <ul style="list-style-type: none"><li>• The count clock is generated by the prescaler. Refer to "6.12 Operating Explanation of Prescaler".</li><li>• Write access to these bits is nullified during timer operation (T00CR1/T01CR1:STA = 1).</li><li>• The clock selection of T01CR0 (timer 01) is nullified during 16-bit operation.</li><li>• These bits cannot be set to "111<sub>B</sub>" when the PWC or input capture function is used. An attempt to write "111<sub>B</sub>" with the PWC or input capture function in use resets the bits to "000<sub>B</sub>". The bits are also reset to "000<sub>B</sub>" if the timer enters the input capture operation mode with the bits set to "111<sub>B</sub>".</li></ul> <table><tr><th>C2</th><th>C1</th><th>C0</th><th>Count clock</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1 × MCLK (machine clock)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1/2 × MCLK (machine clock)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1/4 × MCLK (machine clock)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1/8 × MCLK (machine clock)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1/16 × MCLK (machine clock)</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1/32 × MCLK (machine clock)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1/2<sup>7</sup> × F<sub>CH</sub></td></tr><tr><td>1</td><td>1</td><td>1</td><td>External clock</td></tr></table>	C2	C1	C0	Count clock	0	0	0	1 × MCLK (machine clock)	0	0	1	1/2 × MCLK (machine clock)	0	1	0	1/4 × MCLK (machine clock)	0	1	1	1/8 × MCLK (machine clock)	1	0	0	1/16 × MCLK (machine clock)	1	0	1	1/32 × MCLK (machine clock)	1	1	0	1/2 <sup>7</sup> × F <sub>CH</sub>	1	1	1	External clock
C2	C1	C0	Count clock																																			
0	0	0	1 × MCLK (machine clock)																																			
0	0	1	1/2 × MCLK (machine clock)																																			
0	1	0	1/4 × MCLK (machine clock)																																			
0	1	1	1/8 × MCLK (machine clock)																																			
1	0	0	1/16 × MCLK (machine clock)																																			
1	0	1	1/32 × MCLK (machine clock)																																			
1	1	0	1/2 <sup>7</sup> × F <sub>CH</sub>																																			
1	1	1	External clock																																			

**Table 15.5-1 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) (2 / 2)**

Bit name		Function																																																																																					
bit3 to bit0	F3, F2, F1, F0: Timer operation mode select bits	These bits select the timer operation mode.																																																																																					
		<ul style="list-style-type: none"><li>• The PWM timer function (variable-cycle mode; F3, F2, F1, F0 = 0100<sub>B</sub>) is set by either the T00CR0 (timer 00) register or T01CR0 (timer 01) register. In this case, the other register is set to F3, F2, F1, F0 = 0100<sub>B</sub> automatically when the timer starts operation (T00CR1/T01CR1: STA= 1).</li><li>• The MOD bit is set to "0" automatically when the timer set for 16-bit operation (TMCRO:MOD = 1) starts operation (T00CR1/T01CR1:STA = 1) in the PWM timer function (variable-cycle mode).</li><li>• Write access to these bits is nullified during timer operation (T00CR1/T01CR1:STA = 1).</li></ul>																																																																																					
		<table><tr><th>F3</th><th>F2</th><th>F1</th><th>F0</th><th>Timer operation mode select bits</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Interval timer (one-shot mode)</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Interval timer (continuous mode)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Interval timer (free-run mode)</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>PWM timer (fixed-cycle mode)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>PWM timer (variable-cycle mode)</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>PWC timer ("H" pulse = rising to falling)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>PWC timer ("L" pulse = falling to rising)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>PWC timer (cycle = rising to rising)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>PWC timer (cycle = falling to falling)</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>PWC timer ("H" pulse = rising to falling; Cycle = rising to rising)</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>Input capture (rising, free-run counter)</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Input capture (falling, free-run counter)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Input capture (both edges, free-run counter)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Input capture (rising, counter clear)</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Input capture (falling, counter clear)</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Input capture (both edges, counter clear)</td></tr></table>	F3	F2	F1	F0	Timer operation mode select bits	0	0	0	0	Interval timer (one-shot mode)	0	0	0	1	Interval timer (continuous mode)	0	0	1	0	Interval timer (free-run mode)	0	0	1	1	PWM timer (fixed-cycle mode)	0	1	0	0	PWM timer (variable-cycle mode)	0	1	0	1	PWC timer ("H" pulse = rising to falling)	0	1	1	0	PWC timer ("L" pulse = falling to rising)	0	1	1	1	PWC timer (cycle = rising to rising)	1	0	0	0	PWC timer (cycle = falling to falling)	1	0	0	1	PWC timer ("H" pulse = rising to falling; Cycle = rising to rising)	1	0	1	0	Input capture (rising, free-run counter)	1	0	1	1	Input capture (falling, free-run counter)	1	1	0	0	Input capture (both edges, free-run counter)	1	1	0	1	Input capture (rising, counter clear)	1	1	1	0	Input capture (falling, counter clear)	1	1	1	1	Input capture (both edges, counter clear)
		F3	F2	F1	F0	Timer operation mode select bits																																																																																	
		0	0	0	0	Interval timer (one-shot mode)																																																																																	
		0	0	0	1	Interval timer (continuous mode)																																																																																	
		0	0	1	0	Interval timer (free-run mode)																																																																																	
		0	0	1	1	PWM timer (fixed-cycle mode)																																																																																	
		0	1	0	0	PWM timer (variable-cycle mode)																																																																																	
		0	1	0	1	PWC timer ("H" pulse = rising to falling)																																																																																	
		0	1	1	0	PWC timer ("L" pulse = falling to rising)																																																																																	
		0	1	1	1	PWC timer (cycle = rising to rising)																																																																																	
		1	0	0	0	PWC timer (cycle = falling to falling)																																																																																	
		1	0	0	1	PWC timer ("H" pulse = rising to falling; Cycle = rising to rising)																																																																																	
		1	0	1	0	Input capture (rising, free-run counter)																																																																																	
		1	0	1	1	Input capture (falling, free-run counter)																																																																																	
		1	1	0	0	Input capture (both edges, free-run counter)																																																																																	
		1	1	0	1	Input capture (rising, counter clear)																																																																																	
		1	1	1	0	Input capture (falling, counter clear)																																																																																	
		1	1	1	1	Input capture (both edges, counter clear)																																																																																	

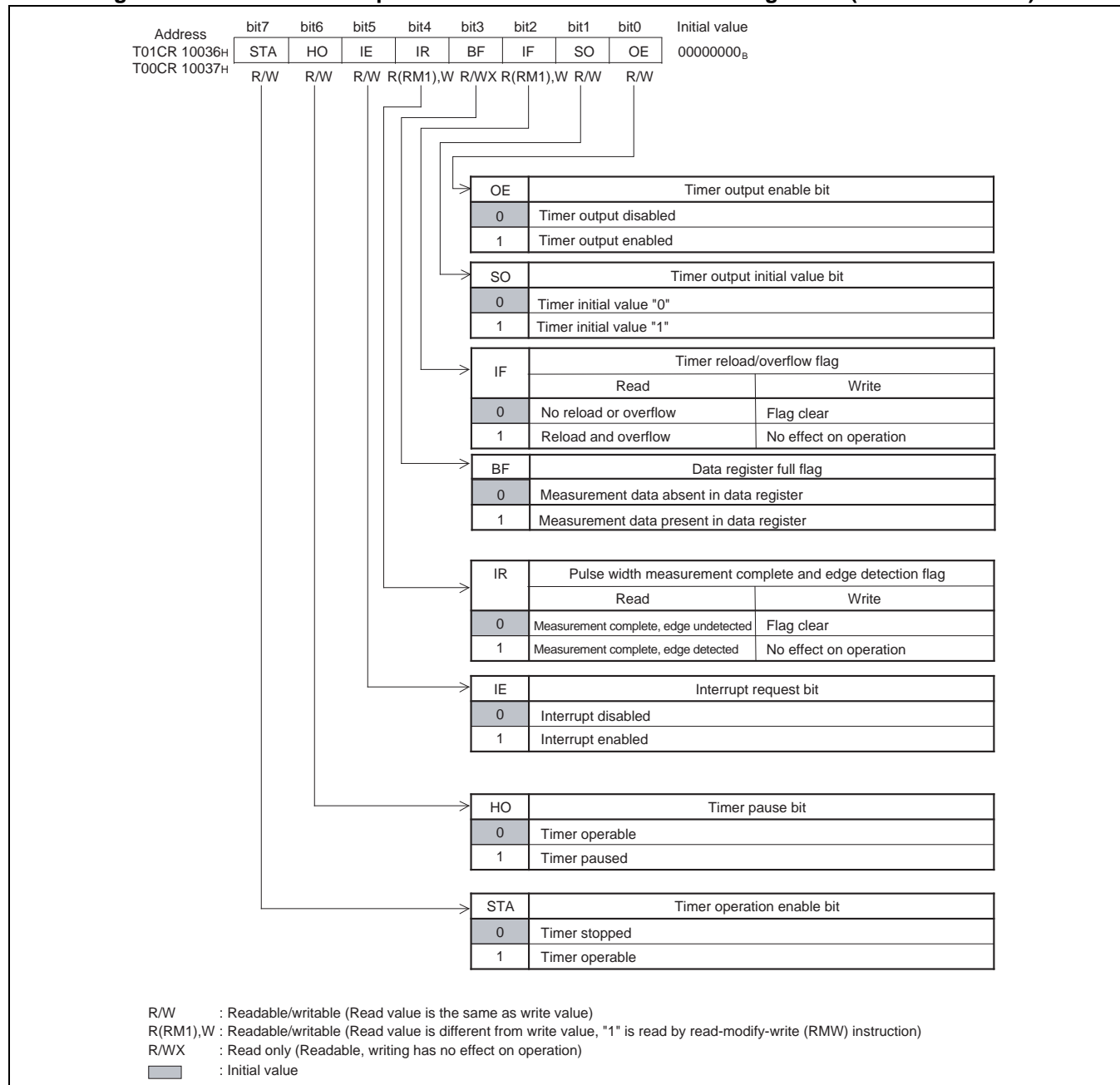


## 15.5.2 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)

8/16-bit compound timer 00/01 control status register 1 (T00CR1/T01CR1) controls the interrupt flag, timer output, and timer operations. T00CR1 and T01CR1 registers correspond to timers 00 and 01, respectively.

### ■ 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)

Figure 15.5-3 8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1)



**Table 15.5-2 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 1 (1 / 2)**

Bit name		Function
bit7	STA: Timer operation enable bit	<p>This bit enables or stops timer operation.</p> <p><b>Writing "0"</b>: stops the timer operation and sets the count value to "00<sub>H</sub>".</p> <ul style="list-style-type: none"> <li>When the PWM timer function (variable-cycle mode) has been selected (T00CR0/T01CR0: F3, F2, F1, F0 = 0100<sub>B</sub>), the STA bit can be used to enable or disable timer operation from within either the T00CR1 (timer 00) or T01CR1 (timer 01) register. In this case, the STA bit in the other register is set to the same value automatically.</li> <li>During 16-bit operation (TMCR0:MOD = 1), use the STA bit in the T00CR1 (timer 00) register to enable or disable timer operation. In this case, the STA bit in the other register is set to the same value automatically.</li> </ul> <p><b>Writing "1"</b>: allows timer operation to start from count value "00<sub>H</sub>".</p> <ul style="list-style-type: none"> <li>Set this bit to "1" after setting the count clock select bits (T00CR0/T01CR0:C2, C1, C0), timer operation select bits (T00CR0/T01CR0:F3, F2, F1, F0), timer output initial value bit (T00CR1/T01CR1:SO), 16-bit mode enable bit (TMCR0:MOD), and filter function select bits (TMCR0:FE11, FE10, FE01, FE00).</li> </ul>
bit6	HO: Timer suspend bit	<p>This bit suspends or resumes timer operation.</p> <ul style="list-style-type: none"> <li>Writing "1" to this bit during timer operation suspends the timer operation.</li> <li>Writing "0" to the bit when timer operation has been enabled (T00CR1/T01CR1:STA = 1) resumes the timer operation.</li> <li>When the PWM timer function (variable-cycle mode) has been selected (T00CR0/T01CR0: F3, F2, F1, F0=0100<sub>B</sub>), the HO bit can be used to suspend or resume timer operation from within either the T00CR1 (timer 00) or T01CR1 (timer 01) register. In this case, the HO bit in the other register is set to the same value automatically.</li> <li>During 16-bit operation (TMCR0:MOD = 1), use the HO bit in the T00CR1 (timer 00) register to suspend or resume timer operation. In this case, the STA bit in the other register is set to the same value automatically.</li> </ul>
bit5	IE: Interrupt request enable bit	<p>This bit enables or disables the output of interrupt requests.</p> <p><b>Writing "0"</b>: disables interrupt request.</p> <p><b>Writing "1"</b>: outputs an interrupt request when the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) or timer reload/overflow flag (T00CR1/T01CR1:IF) is "1".</p> <p>Note, however, that an interrupt request from the timer reload/overflow flag (T00CR1/T01CR1:IF) is not outputted unless the IF flag interrupt enable (T00CR0/T01CR0:IFE) bit is also set to "1".</p>
bit4	IR: Pulse width measurement completion/edge detection flag	<p>This bit shows the completion of pulse width measurement or the detection of an edge.</p> <ul style="list-style-type: none"> <li>The bit is set to "1" upon completion of pulse width measurement when the PWC timer function has been selected.</li> <li>The bit is set to "1" upon detection of an edge when the input capture function has been selected.</li> <li>The bit is "0" when any timer function other than the PWC timer and input capture functions has been selected.</li> <li>This bit always returns "1" to a read modify write (RMW) instruction.</li> <li>The IR bit in T01CR1 (timer 01) register is set to "0" during 16-bit operation.</li> <li>Writing "0" to the bit sets it to "0".</li> <li>An attempt to write "1" to the bit is ignored.</li> </ul>

**Table 15.5-2 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Control Status Register 1 (2 / 2)**

Bit name		Function
bit3	BF: Data register full flag	<ul style="list-style-type: none"> <li>This bit is set to "1" when a count value is stored in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) upon completion of pulse width measurement in PWC timer function.</li> <li>This bit is set to "0" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is read during 8-bit operation.</li> <li>The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) holds data with this bit containing "1". Even when the next edge is detected with this bit containing "1", the count value is not transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and thus the next measurement result is lost. However, as the exception, when the "H" pulse and cycle measurement (T00CR0/T01CR0: F3, F2, F1, F0= 1001<sub>B</sub>) is selected, the "H" pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) with this bit set to "1". The cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the bit set to "1". For cycle measurement, therefore, the "H" pulse measurement result must be read before the cycle is completed. Note also that the result of "H" pulse measurement or cycle measurement is lost unless read before the completion of the next "H" pulse.</li> <li>The BF bit in the T00CR1 (timer 00) register is set to "0" when the T01DR (timer 01) register is read during 16-bit operation.</li> <li>The BF bit in T01CR1 (timer 01) register is set to "0" during 16-bit operation.</li> <li>This bit is "0" when any timer function other than the PWC timer function has been selected.</li> <li>Writing to this bit has no effects on the operation.</li> </ul>
bit2	IF: Timer reload/overflow flag	<p>This bit detects a match with a count value or a counter overflow.</p> <ul style="list-style-type: none"> <li>The bit is set to "1" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) value matches the count value during interval timer function (both one-shot and continuous mode) or PWM timer function (variable-cycle mode).</li> <li>The bit is set to "1" when a counter overflow occurs during PWC or input capture function.</li> <li>This bit always returns "1" to a read-modify-write (RMW) instruction.</li> <li>Writing "0" to the bit sets it to "0".</li> <li>Writing "1" to this bit has no effects on the operation.</li> <li>The bit is "0" when the PWM function (variable-cycle mode) has been selected.</li> <li>The IF bit in the T01CR1 (timer 01) register is "0" during 16-bit operation.</li> </ul>
bit1	SO: Timer output initial value bit	<p>Writing to this bit sets the timer output (TMCR0:TO1/TO0) initial value. The value in this bit is reflected in the timer output when the timer operation enable bit (T00CR1/T01CR1:STA) changes from "0" to "1".</p> <ul style="list-style-type: none"> <li>During 16-bit operation (TMCR0:MOD = 1), use the SO bit in the T00CR1 (timer 00) register to set the timer output initial value. In this case, the value of the S bit in the other register is meaningless.</li> <li>An attempt to write to this bit is nullified during timer operation (T00CR1/T01CR1:STA = 1). During 16-bit operation, however, a value can be written to the SO bit in the T01CR1 (timer 01) register even during timer operation but it has no direct effect on the timer output.</li> <li>The value of this bit is meaningless when the PWM timer function (either fixed-cycle or variable-cycle mode) or input capture function has been selected.</li> </ul>
bit0	OE: Timer output enable bit	<p>This bit enables or disabled timer output.</p> <p><b>Writing "0"</b>: prevents the timer output from being supplied to the external pin. In this case, the external pin serves as a general-purpose port.</p> <p><b>Writing "1"</b>: supplies timer output (TMCR0:TO1/TO0) to the external pin.</p>

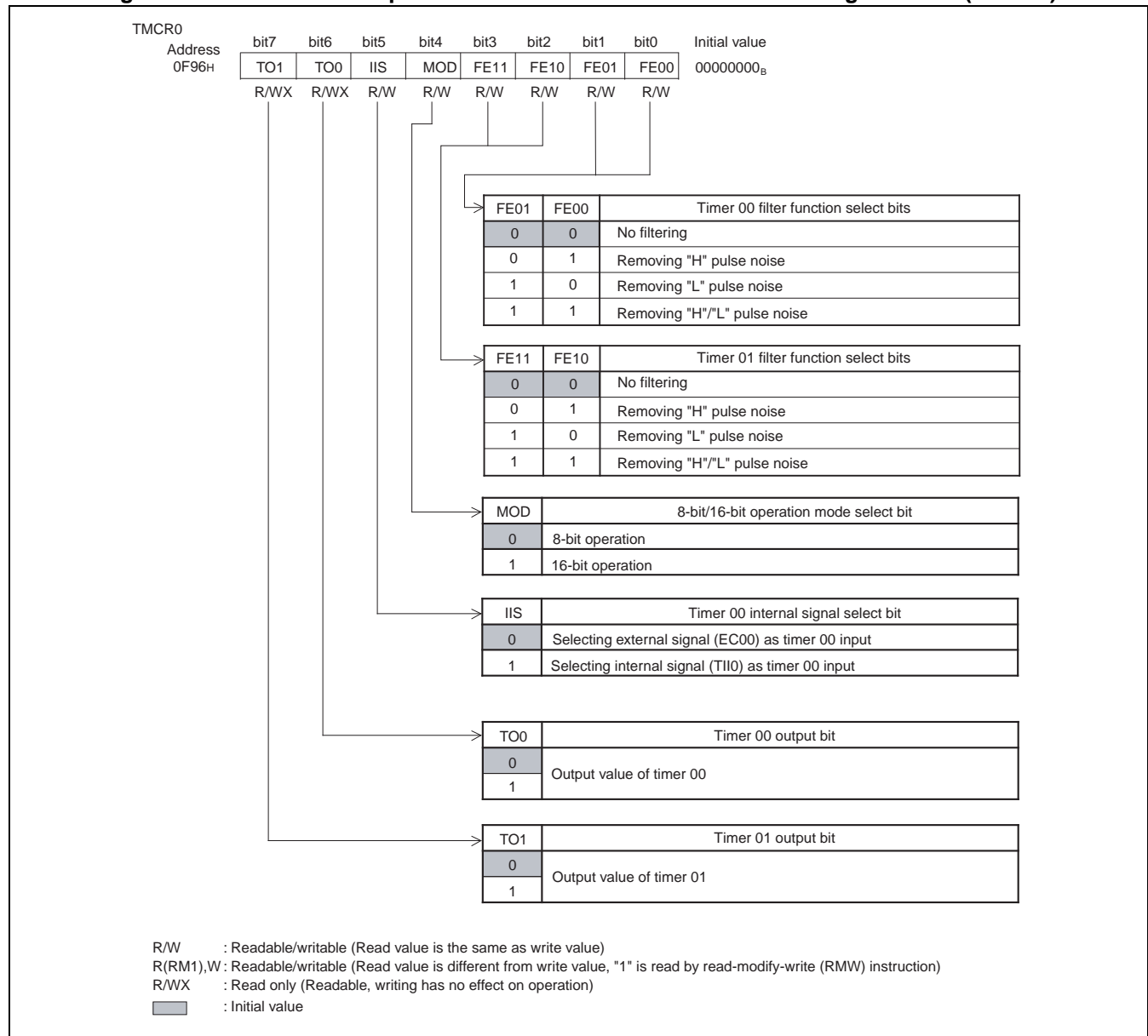
## MB95110B/M Series

### 15.5.3 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

The 8/16-bit compound timer 00/01 timer mode control register ch.0 (TMCR0) selects the filter function, 8-bit or 16-bit operation mode, and signal input to timer 00 and to indicate the timer output value. This register serves for both of timers 00 and 01.

#### ■ 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)

Figure 15.5-4 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0)



**Table 15.5-3 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) (1 / 2)**

Bit name		Function															
bit7	TO1: Timer 01 output bit	<p>This bit indicates the output value of timer 01. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value in the bit changes depending on the selected timer function.</p> <ul style="list-style-type: none"> <li>Writing to this bit has no effect on the operation.</li> <li>The value in the bit remains indeterminate during 16-bit operation when the PWM timer function (variable-cycle mode) or input capture function has been selected.</li> <li>When the timer stops operation (T00CR1/T01CR1:STA = 0) in interval timer or PWC timer function, this bit holds the last value.</li> <li>When the timer stops operation in PWM timer function (fixed-cycle mode), this bit holds the last value.</li> <li>When the timer operation mode select bit (T00CR0/T01CR0: F3, F2, F1, F0) is changed with the timer being stopped, the bit indicates the last value of timer operation if the same timer operation has ever been performed or otherwise contains "0".</li> </ul>															
bit6	TO0: Timer 00 output bit	<p>This bit indicates the output value of timer 00. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value in the bit changes depending on the selected timer function.</p> <ul style="list-style-type: none"> <li>Writing to this bit has no effect on the operation.</li> <li>The value in the bit remains indeterminate when the input capture function has been selected.</li> <li>When the timer stops operation (T00CR1/T01CR1:STA = 0) in interval timer, PWM timer (variable-cycle mode), or PWC timer function, this bit holds the last value.</li> <li>When the timer stops operation in PWM timer function (fixed-cycle mode), this bit holds the last value.</li> <li>When the timer operation mode select bit (T00CR0/T01CR0: F3, F2, F1, F0) is changed with the timer being stopped, the bit indicates the last value of timer operation if the same timer operation has ever been performed or otherwise contains "0".</li> </ul>															
bit5	IIS: Timer 00 internal signal select bit	<p>This bit selects the signal input to timer 00 when the PWC timer or input capture function has been selected.</p> <p><b>Writing "0"</b>: selects the external signal (EC00) as the signal input for timer 00.</p> <p><b>Writing "1"</b>: selects the internal signal (THI0) as the signal input for timer 00.</p>															
bit4	MOD: 16-bit mode enable bit	<p>This bit selects 8-bit or 16-bit operation mode.</p> <p><b>Writing "0"</b>: allows timers 00 and 01 to operate as separate 8-bit timers.</p> <p><b>Writing "1"</b>: allows timers 00 and 01 to operate as a 16-bit timer.</p> <ul style="list-style-type: none"> <li>This bit is set to "0" automatically when the timer starts operation (T00CR1/T01CR1:STA=1) in PWM timer mode (variable-cycle mode).</li> <li>Write access to this bit is nullified during timer operation (T00CR1:STA = 1 or T01CR1:STA = 1).</li> </ul>															
bit3, bit2	FE11, FE10: Timer 01 filter function select bits	<p>These bits select the filter function for the external signal (EC01) to timer 01 when the PWC timer or input capture function has been selected.</p> <table border="1"> <thead> <tr> <th>FE11</th><th>FE10</th><th>Timer 01 filter function</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>No filtering</td></tr> <tr> <td>0</td><td>1</td><td>Removing "H" pulse noise</td></tr> <tr> <td>1</td><td>0</td><td>Removing "L" pulse noise</td></tr> <tr> <td>1</td><td>1</td><td>Removing "H"/"L" pulse noise</td></tr> </tbody> </table> <ul style="list-style-type: none"> <li>Write access to these bits is nullified during timer operation (T01CR1:STA = 1).</li> <li>The settings of the bits have no effect on operation when the interval timer or PWM timer function has been selected (filter function does not operate.).</li> </ul>	FE11	FE10	Timer 01 filter function	0	0	No filtering	0	1	Removing "H" pulse noise	1	0	Removing "L" pulse noise	1	1	Removing "H"/"L" pulse noise
FE11	FE10	Timer 01 filter function															
0	0	No filtering															
0	1	Removing "H" pulse noise															
1	0	Removing "L" pulse noise															
1	1	Removing "H"/"L" pulse noise															

**Table 15.5-3 Functional Description of Each Bit of 8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) (2 / 2)**

Bit name		Function		
bit1, bit0	FE01, FE00: Timer 00 filter function select bits	These bits select the filter function for the external signal (EC00) to timer 00 when the PWC timer or input capture function has been selected.		
		FE01	FE00	Timer 00 filter function
		0	0	No filtering
		0	1	Removing "H" pulse noise
		1	0	Removing "L" pulse noise
		1	1	Removing "H"/"L" pulse noise
		<ul style="list-style-type: none"><li>• An attempt to write to these bits is nullified during timer operation (T00CR1:STA = 1).</li><li>• The settings of these bits have no effect on operation when the interval timer or PWM timer function has been selected (filter function does not operate.).</li></ul>		

## 15.5.4 8/16-bit Compound Timer 00/01 Data Register ch.0 (T00DR/T01DR)

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to write the maximum value counted during interval timer or PWM timer operation and to read the count value during PWC timer or input capture operation. The T00DR and T01DR registers correspond to timers 00 and 01, respectively.

### ■ 8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR)

Figure 15.5-5 8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
T01DR 0F94 <sub>H</sub>	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	00000000 <sub>B</sub>
T00DR 0F95 <sub>H</sub>	R,W	R,W	R,W	R,W	R,W	R,W	R,W	R,W	

R,W: Readable, writable (Read value is different from write value)

#### ● Interval timer function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to set the interval time. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value of this register is transferred to the latch in the 8-bit comparator and the counter starts counting. When the count value matches the value held in the latch in the 8-bit comparator, the value of this register is transferred again to the latch and the count value is reset to "00<sub>H</sub>" to continue to count.

The current count value can be read from this register.

An attempt to write "00<sub>H</sub>" to this register is disabled in interval timer function.

In 16-bit operation, set the upper data to T01DR and lower data to T00DR. And, write and read T01DR and T00DR in this order.

#### ● PWM timer functions (fixed-cycle)

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to set "H" pulse width time. When the timer starts operation (T00CR1/T01CR1:STA=1), the value of this register is transferred to the latch in the 8-bit comparator and the counter starts counting from timer output "H". When the count value matches the value held in the latch, the timer output becomes "L" and the counter continues to count until the count value reaches "FF<sub>H</sub>". When an overflow occurs, the value of this register is transferred again to the latch in the 8-bit comparator and the counter performs the next cycle of counting.

The current value can be read from this register. In 16-bit operation, set the upper data to T01DR and lower data to T00DR. And, write and read T01DR and T00DR in this order.

● PWM timer functions (variable-cycle)

The 8/16-bit compound timer 00 data register (T00DR) and 8/16-bit compound timer 01 data register (T01DR) are used to set "L" pulse width timer and cycle, respectively. When the timer starts operation (T00CR1/T01CR1:STA = 1), the value of each register is transferred to the latch in the 8-bit comparator and two counters start counting from timer output "L". When the T00DR value held in the latch matches the timer 00 counter value, the timer output becomes "H" and the counting continues until the T01DR value held in the latch matches the timer 01 counter value. When the T01DR value held in the latch of the 8-bit comparator matches the timer 01 counter value, the values of these registers are transferred again to the latch and the next PWM cycle of counting is performed continuously.

The current count value can be read from this register.

In 16-bit operation, set the upper data and lower data to T01DR and T00DR, respectively. And, write and read T01DR and T00DR in this order.

● PWC timer function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to read PWC measurement results. When PWC measurement is completed, the counter value is transferred to this register and the BF bit is set to "1".

When the 8/16-bit compound timer 00/01 data register is read, the BF bit is set to "0". Transfer to the 8/16-bit compound timer 00/01 data register is not performed with the BF bit containing "1".

As the exception, when the "H" pulse and cycle measurement (T00CR0/T01CR0:F3, F2, F1, F0 = 1001<sub>B</sub>) is selected, the "H" pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1", but the cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1". For cycle measurement, therefore, the "H" pulse measurement result must be read before the cycle is completed. Note also that the result of "H" pulse measurement or cycle measurement is lost unless read before the completion of the next "H" pulse.

When reading the 8/16-bit compound timer 00/01 data register, be careful not to clear the BF bit unintentionally.

Writing to the 8/16-bit compound timer 00/01 data register updates the stored measurement data with the write value. Therefore, do not perform a write operation. In 16-bit operation, the upper data and lower data are transferred to T01DR and T00DR, respectively. Read T01DR and T00DR in this order.

● Input capture function

The 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is used to read input capture results. When a specified edge is detected, the counter value is transferred to the 8/16-bit compound timer 00/01 data register.

Writing a value to the data register updates the measurement data stored there with that value. Therefore, do not write to the 8/16-bit compound timer 00/01 data register. In 16-bit operation, the upper data and lower data are transferred to T01DR and T00DR, respectively. Read T01DR and T00DR in this order.



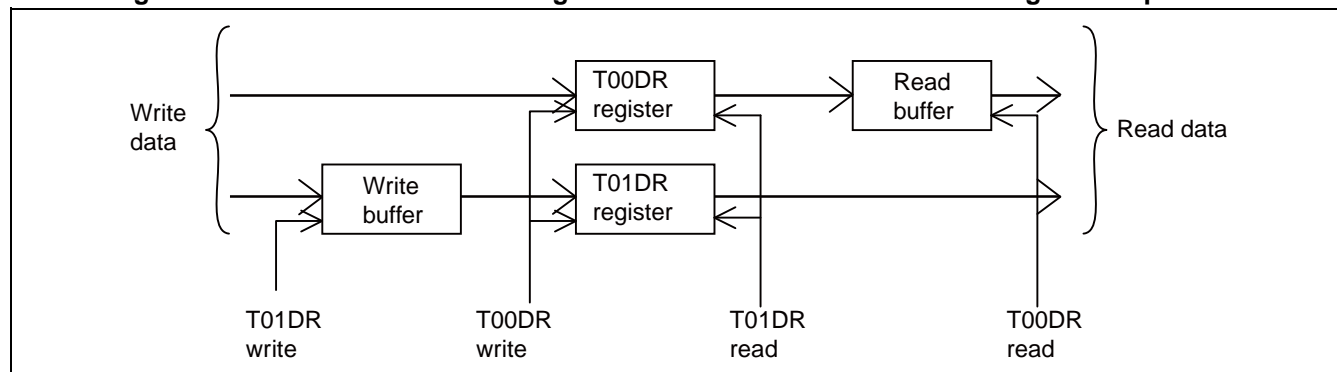
● Read and write operations

Read and write operations of T00DR and T01DR are performed in the following manner during 16-bit operation and PWM timer function (variable-cycle).

- Read from T01DR: Read access from the register also involves storing the T00DR value into the internal read buffer.
- Read from T00DR: Read from the internal read buffer.
- Write to T01DR: Write to the internal write buffer.
- Write to T00DR: Write access to the register also involves storing the value of the internal write buffer into T01DR.

Figure 15.5-6 shows the T00DR and T01DR registers read from and written to during 16-bit operation.

**Figure 15.5-6 T00DR and T01DR registers read from and written to during 16-bit operation**



## 15.6 Interrupts of 8/16-bit Compound Timer

The 8/16-bit compound timer generates the following types of interrupts to each of which an interrupt number and interrupt vector are assigned.

- Timer 00 interrupt
- Timer 01 interrupt

### ■ Timer 00 Interrupt

Table 15.6-1 explains the timer 00 interrupt and its source.

**Table 15.6-1 Timer 00 Interrupt**

Item	Description		
Interrupt generating condition	Comparison match in interval timer function or PWM timer function (variable-cycle mode) has been selected	Overflow in PWC timer function or input capture function	Completion of measurement in PWC timer function or edge detection in input capture function
Interrupt flag	T00CR1:IF	T00CR1:IF	T00CR1:IR
Interrupt enable	T00CR1:IE and T00CR0:IFE	T00CR1:IE and T00CR0:IFE	T00CR1:IE

### ■ Timer 01 Interrupt

Table 15.6-2 explains the timer 01 interrupt and its cause.

**Table 15.6-2 Timer 01 Interrupt**

Item	Description		
Interrupt generating condition	Comparison match in interval timer function or PWM timer function (variable-cycle mode) has been selected Excluded during 16-bit operation	Overflow in PWC timer function or input capture function Excluded during 16-bit operation	Completion of measurement in PWC timer function or edge detection in input capture function Excluded during 16-bit operation
Interrupt flag	T01CR1:IF	T01CR1:IF	T01CR1:IR
Interrupt enable	T01CR1:IE and T01CR0:IFE	T01CR1:IE and T01CR0:IFE	T01CR1:IE

■ **Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer**

**Table 15.6-3 Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer**

Interrupt source	Interrupt request No.	Interrupt level setup register		Vector table address	
		Register	Setting bit	Upper	Lower
Timer 00	IRQ5	ILR1	L05	FFF0 <sub>H</sub>	FFF1 <sub>H</sub>
Timer 01	IRQ6	ILR1	L06	FFEE <sub>H</sub>	FFEF <sub>H</sub>
Timer 10 <sup>*</sup>	IRQ22	ILR5	L22	FFCE <sub>H</sub>	FFCF <sub>H</sub>
Timer 11	IRQ14	ILR3	L14	FFDE <sub>H</sub>	FFDF <sub>H</sub>

\*: 8/16-bit compound timer (ch.1) shares the same interrupt request number and vector table as the external interrupt circuit (ch.12 to ch.15).

The request numbers and vector tables of all peripheral functions are listed in Appendix B "Table of Interrupt Causes".

## 15.7 Operating Description of Interval Timer Function (One-shot Mode)

This section describes the operations of the interval timer function (one-shot mode) for the 8/16-bit compound timer.

### ■ Operation of Interval Timer Function (One-shot Mode)

The compound timer requires the register settings shown in Figure 15.7-1 to serve as the interval timer function.

**Figure 15.7-1 Settings of Interval Timer Function**

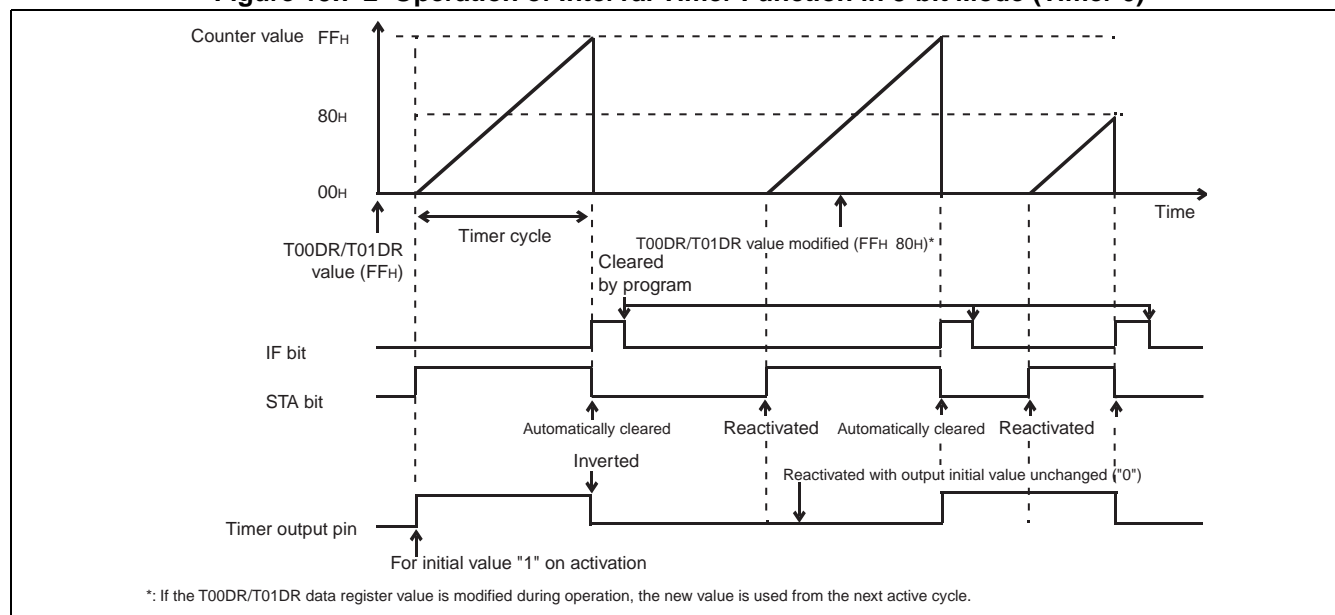
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	0	0	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	x	x	○	○	○
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	x	○	○	○	○	○
T00DR/T01DR	Sets interval timer (counter compare value)							
	○: Used bit							
	x: Unused bit							
	1: Set "1"							
	0: Set "0"							

In interval timer function (one-shot mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00<sub>H</sub>" at the rising edge of a selected count clock signal. When the counter value matches the value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1" and the start bit (T00CR0/T00CR1:STA) is set to "0", and then the count operation stops.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator when the counter starts counting. Writing "00<sub>H</sub>" to the 8/16-bit compound timer 00/01 data register is prohibited.

Figure 15.7-2 shows the operation of the interval timer function in the 8-bit operation.

Figure 15.7-2 Operation of Interval Timer Function in 8-bit Mode (Timer 0)



## 15.8 Operating Description of Interval Timer Function (Continuous Mode)

This section describes the interval timer function (continuous mode operation) of the 8/16-bit compound timer.

### ■ Operation of Interval Timer Function (Continuous Mode)

The compound timer requires the register settings shown in Figure 15.8-1 to serve as the interval timer function (continuous mode).

**Figure 15.8-1 Settings for Counter Function (8-bit Mode)**

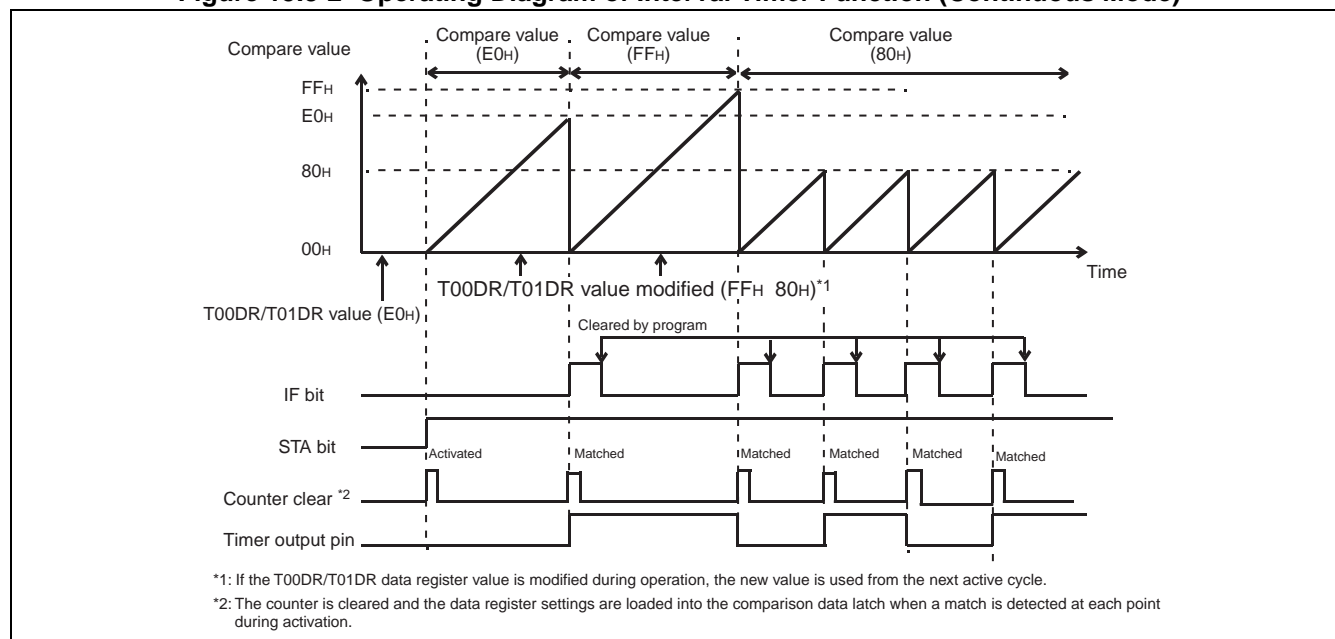
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	0	0	1
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	x	x	○	○	○
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	x	○	○	○	○	○
T00DR/T01DR	Sets interval time (counter compare value)							
	○: Used bit							
	x: Unused bit							
	1: Set "1"							
	0: Set "0"							

In interval timer function (continuous mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00<sub>H</sub>" at the rising edge of a selected count clock signal. When the counter value matches the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted, the interrupt flag (T00CR1/T01CR1:IF) is set to "1", and the counter continues to count by restarting at "00<sub>H</sub>". The timer outputs a square wave as a result of this continuous operation.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected. Writing "00<sub>H</sub>" to the 8/16-bit compound timer 00/01 data register is disabled during the count operation.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

Figure 15.8-2 Operating Diagram of Interval Timer Function (Continuous Mode)



## 15.9 Operating Description of Interval Timer Function (Free-run Mode)

This section describes the operation of the interval timer function (free-run mode) for the 8/16-bit compound timer.

### ■ Operation of Interval Timer Function (Free-run Mode)

The compound timer requires the settings shown in Figure 15.9-1 to serve as the interval timer function (free-run mode).

**Figure 15.9-1 Settings for Interval Timer Function (Free-run Mode)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	0	1	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	x	x	○	○	○
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	x	○	○	○	○	○
T00DR/T01DR	Sets interval time (counter compare value)							
	○: Used bit							
	x: Unused bit							
	1: Set "1"							
	0: Set "0"							

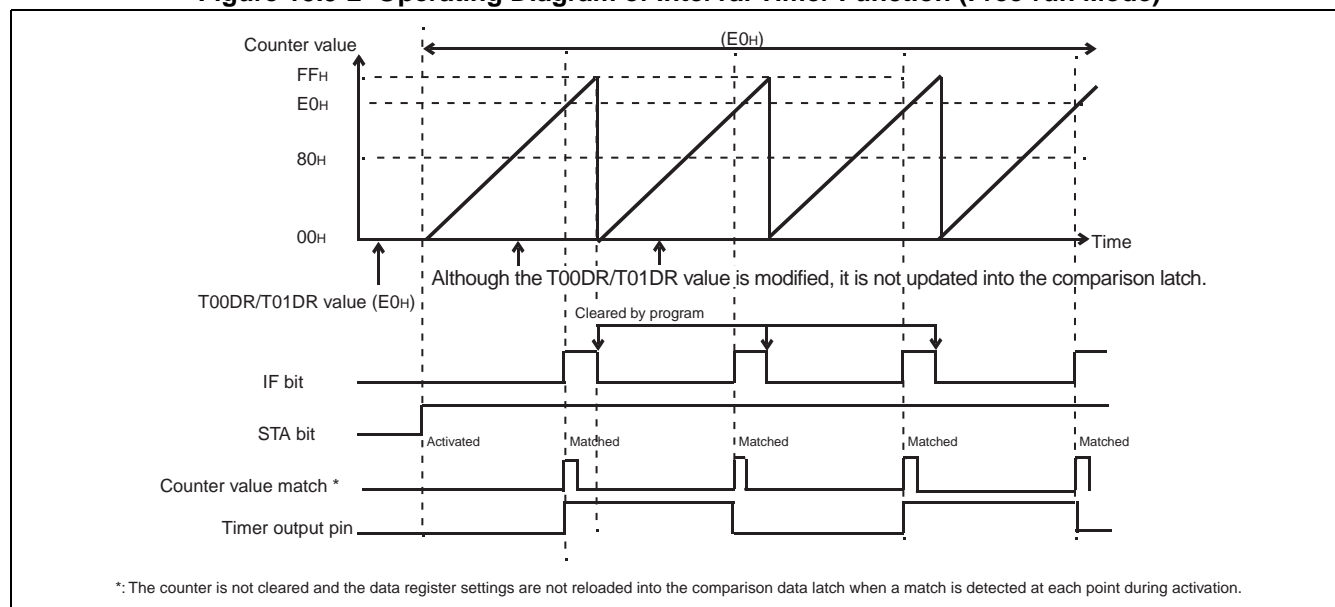
In interval timer function (free-run mode), enabling timer operation (T00CR0/T00CR1:STA = 1) causes the counter to start counting from "00<sub>H</sub>" at the rising edge of a selected count clock signal. When the counter value matches the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the timer output bit (TMCR0:TO0/TO1) is inverted and the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The counter continues to count, and when the count value reaches "FF<sub>H</sub>", it restarts counting at "00<sub>H</sub>" to continue. The timer outputs a square wave as a result of this continuous operation.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected. Writing "00<sub>H</sub>" to the 8/16-bit compound timer 00/01 data register is prohibited.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.



Figure 15.9-2 Operating Diagram of Interval Timer Function (Free-run Mode)



## 15.10 Operating Description of PWM Timer Function (Fixed-cycle mode)

This section describes the operation of the PWM timer function (fixed-cycle mode) for the 8/16-bit compound timer.

### ■ Operation of PWM Timer Function (Fixed-cycle Mode)

The compound timer requires the settings shown in Figure 15.10-1 to serve as the PWM timer function (fixed-cycle mode).

**Figure 15.10-1 Settings for PWM Timer Function (Fixed-cycle Mode)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	0	1	1
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	x	x	x	x	x	x
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	x	○	○	○	○	○
T00DR/T01DR	Sets "H" pulse width (compare value)							
	○: Used bit							
	x: Unused bit							
	1: Set "1"							
	0: Set "0"							

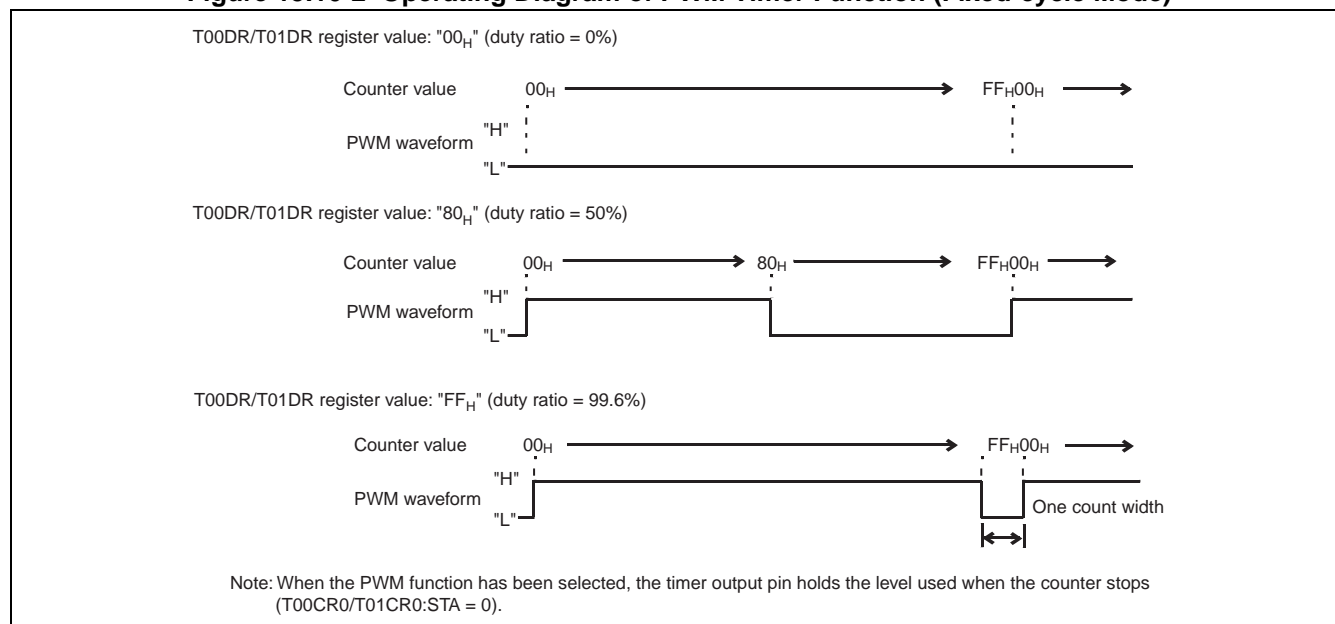
In PWM timer function (fixed-cycle mode), a fixed cycle PWM signal in a variable "H" pulse width is outputted from the timer output pin (TO00/TO01). The cycle is fixed to "FF<sub>H</sub>" in 8-bit operation or "FFFF<sub>H</sub>" in 16-bit operation. The time is determined by the count clock selected. The "H" pulse width is specified by the value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR).

This function has no effect on the interrupt flag (T00CR1/T01CR1:IF). As each cycle always starts with "H" pulse output, the timer output initial value setting bit (T00CR1/T01CR1:SO) is meaningless.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a counter value comparison match is detected.

When the timer stops operation, the timer output bit (TMCR0:TO0/TO1) holds the last value.

The "H" pulse is one count clock shorter than the setting value in the output waveform immediately after activation of the timer (write "1" to the STA bit).

**Figure 15.10-2 Operating Diagram of PWM Timer Function (Fixed-cycle Mode)**

## 15.11 Operating Description of PWM Timer Function (Variable-cycle Mode)

This section describes the operations of the PWM timer function (variable-cycle mode) for the 8/16-bit compound timer.

### ■ Operation of PWM Timer Function (Variable-cycle Mode)

The compound timer requires the settings shown in Figure 15.11-1 to serve as the PWM timer function (variable-cycle mode).

**Figure 15.11-1 Settings for PWM Timer Function (Variable-cycle Mode)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	0	1	0	0
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	x	x	○	x	x
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	x	x	○	○	○	○
T00DR	Sets "L" pulse width (compare value)							
T01DR	Sets the cycle of PWM waveform (compare value)							
○: Used bit								
x: Unused bit								
1: Set "1"								
0: Set "0"								

In PWM timer function (variable-cycle mode), both timers 00 and 01 are used when the cycle is specified by the 8/16-bit compound timer 01 data register (T01DR), and the "L" pulse width is specified by the 8/16-bit compound timer 00 data register (T00DR), any cycle and duty PWM signal is generated from the timer output bit (TO00).

For this function, the compound timer cannot serve as a 16-bit counter as the two 8-bit counters are used.

Enabling timer operation (by setting either T00CR1:STA = 1 or T01CR1:STA = 1) sets the mode bit (TMCR0:MOD) to "0". As the first cycle always begins with "L" pulse output, the timer initial value setting bit (T00CR1/T01CR1:SO) is meaningless.

The interrupt flag (T00CR1/T01CR1:IF) is set when each 8-bit counter matches the value in the corresponding 8/16-bit compound timer 00/01 data register (T00DR/T01DR).

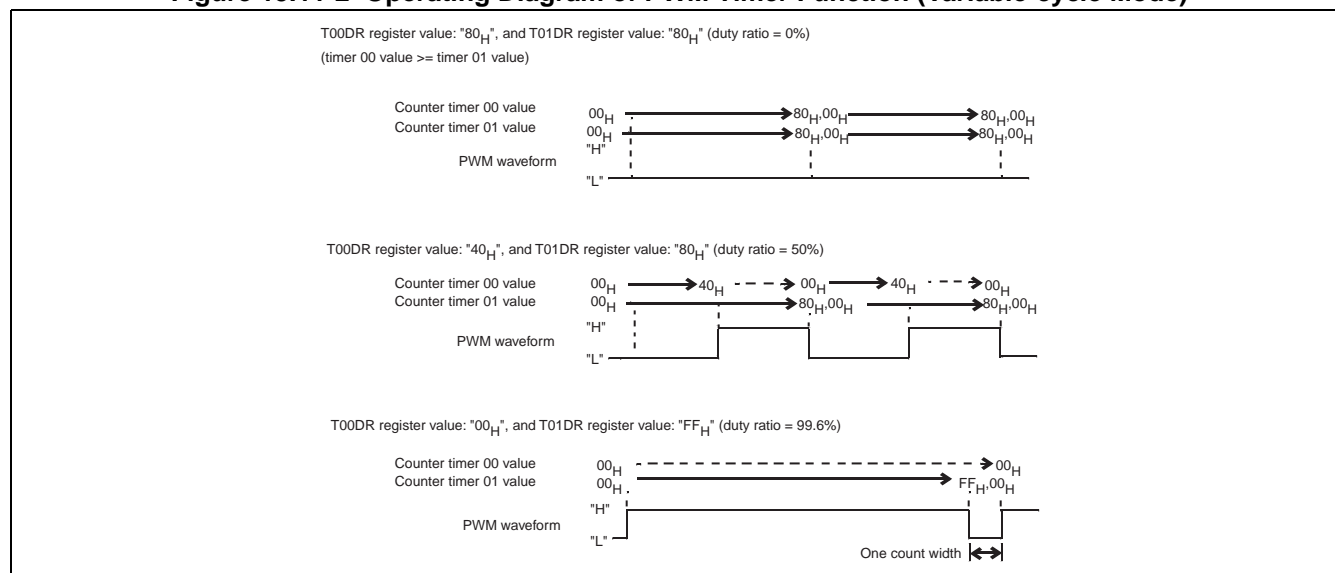
The 8/16-bit compound timer 00/01 data register value is transferred to the temporary storage latch (comparison data storage latch) in the comparator either when the counter starts counting or when a comparison match with each counter value is detected.

"H" is not outputted when the "L" pulse width setting value is greater than the cycle setting value.

The count clock must be selected for both of timers 00 and 01. Selecting different count clocks, however, is prohibited.

When the timer stops operation, the timer output bit (TMCR0:TO0) holds the last output value.

If the 8/16-bit compound timer 00/01 data register is written over during operation, the written data will be effective from the cycle immediately after the detection of a synchronous match.

**Figure 15.11-2 Operating Diagram of PWM Timer Function (Variable-cycle Mode)**

## 15.12 Operating Description of PWC Timer Function

This section describes the operations of the PWC timer function for the 8/16-bit compound timer.

### ■ Operation of PWC Timer Function

The compound timer requires the settings shown in Figure 15.12-1 to serve as the PWC timer function.

**Figure 15.12-1 Settings for PWC Timer Function**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	○	○	○	○
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	○	○	○	○	x
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	○	○	○	○	○	○	○	○
T00DR/T01DR	Holds pulse width measurement value							

○: Used bit  
x: Unused bit  
1: Set "1"

When the PWC timer function is selected, the width and cycle of an external input pulse can be measured. The edges to start and end counting are selected by timer operation mode setting (T00CR0/T01CR0:F3, F2, F1, F0).

In this operation mode, the counter starts counting from "00<sub>H</sub>" upon detection of the specified count start edge of an external input signal. Upon detection of the specified count end edge, the count value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) and buffer full flag (T00CR1/T01CR1:BF) are set to "1". The buffer full flag is set to "0" when the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) is read from.

The 8/16-bit compound timer 00/01 data register holds data with the buffer full flag set to "1". Even when the next edge is detected at this time, the next measurement result is lost as the count value is not transferred to the 8/16-bit compound timer 00/01 data register.

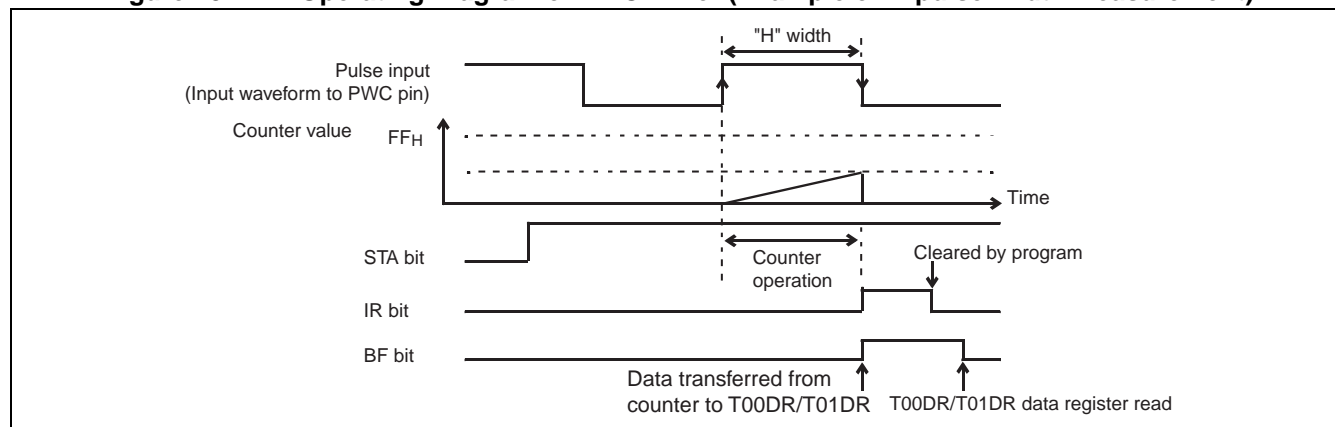
As the exception, when the H-pulse and cycle measurement (T00CR0/T01CR0:F3, F2, F1, F0 = "1001<sub>B</sub>") is selected, the H-pulse measurement result is transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1", but the cycle measurement result is not transferred to the 8/16-bit compound timer 00/01 data register with the BF bit set to "1". For cycle measurement, therefore, the H-pulse measurement result must be read before the cycle is completed. Note also that the result of H-pulse measurement or cycle measurement is lost unless read before the completion of the next H pulse.

To measure the time exceeding the counter value, you can use software to count the number of occurrences of a counter overflow. When the counter causes an overflow, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of times the overflow occurs. Note also that an overflow toggles the timer output. The timer output initial value can be set by the timer output initial value bit (T00CR1/T01CR1:SO).

When the timer stops operation, the timer output bit (TMCR0:TO1/TO0) holds the last value.

The value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) must be nullified if an interrupt occurs before the timer is activated (before "1" is written to the STA bit).

**Figure 15.12-2 Operating Diagram of PWC Timer (Example of H-pulse Width Measurement)**



## 15.13 Operating Description of Input Capture Function

This section describes the operations of the input capture function for the 8/16-bit compound timer.

### ■ Operation of Input Capture Function

The compound timer requires the settings shown in Figure 15.13-1 to serve as the input capture function.

**Figure 15.13-1 Settings for Input Capture Function**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
T00CR0/T01CR0	IFE	C2	C1	C0	F3	F2	F1	F0
	○	○	○	○	○	○	○	○
T00CR1/T01CR1	STA	HO	IE	IR	BF	IF	SO	OE
	1	○	○	○	x	○	x	x
TMCR0	TO1	TO0	IIS	MOD	FE11	FE10	FE01	FE00
	x	x	○	○	○	○	○	○
T00DR/T01DR	Holds pulse width measurement value							
	○: Used bit							
	x: Unused bit							
	1: Set "1"							

When the input capture function is selected, the counter value is stored to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) upon detection of an edge of the external signal input. The edge to be detected is selected by timer operation mode setting (T00CR0/T01CR0:F3, F2, F1, F0).

This function is available in either free-run mode or clear mode, which can be selected by timer operation mode setting.

In clear mode, the counter starts counting from "00<sub>H</sub>". When the edge is detected, the counter value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR), the interrupt flag (T00CR1/T01CR1:IR) is set to "1", and the counter continues to count by restarting at "00<sub>H</sub>".

When the edge is detected in free-run mode, the counter value is transferred to the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) and the interrupt flag (T00CR1/T01CR1:IR) is set to "1". In this case, the counter continues to count without being cleared.

This function has no effect on the buffer full flag (T00CR1/T01CR1:BF).

To measure the time exceeding the counter value, software can be used to count the number of occurrences of a counter overflow. When the counter causes an overflow, the interrupt flag (T00CR1/T01CR1:IF) is set to "1". The interrupt service routine can therefore be used to count the number of times the overflow occurs.

The capture value in the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) must be nullified if an interrupt occurs before the timer is activated (before "1" is written to the STA bit).



When the timing at which the 8/16-bit compound timer captures a counter value is the detection of either edge of the external input signal (T00CR0/T01CR0:F3-F0=1100<sub>B</sub> or 1111<sub>B</sub>), the operations in falling edge detection vary according to the level of the external input signal as explained below.

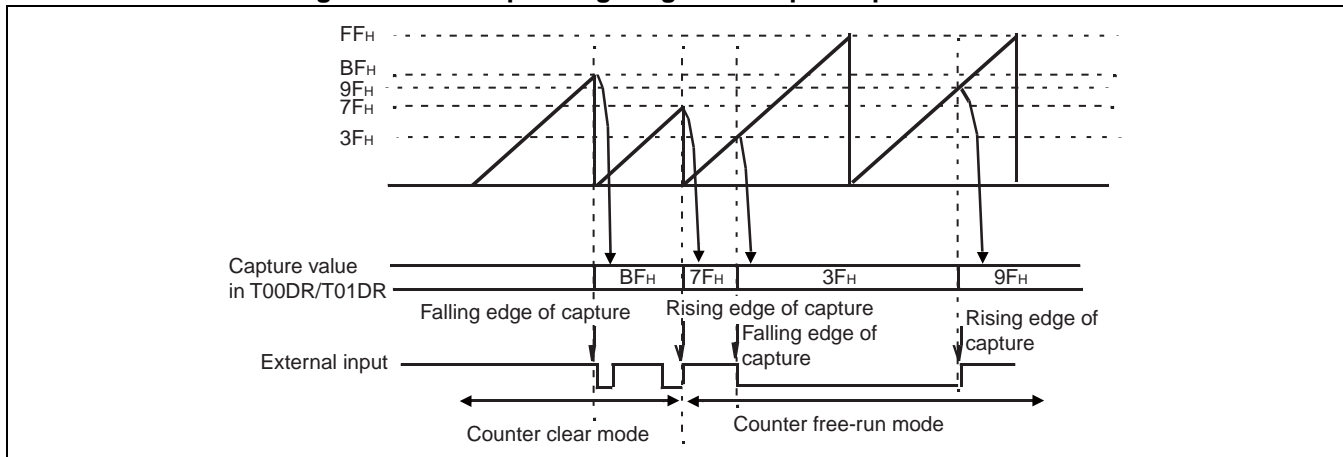
- External input signal level: H

In both free-run mode and clear mode, the first falling edge is ignored, no counter value is transferred to the data register (T00DR/T01DR), and the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) is not set. In addition, in clear mode, the counter is not cleared either.

- External input signal level: L

The 8/16-bit compound timer starts edge detection from the first rising edge.

**Figure 15.13-2 Operating Diagram of Input Capture Function**

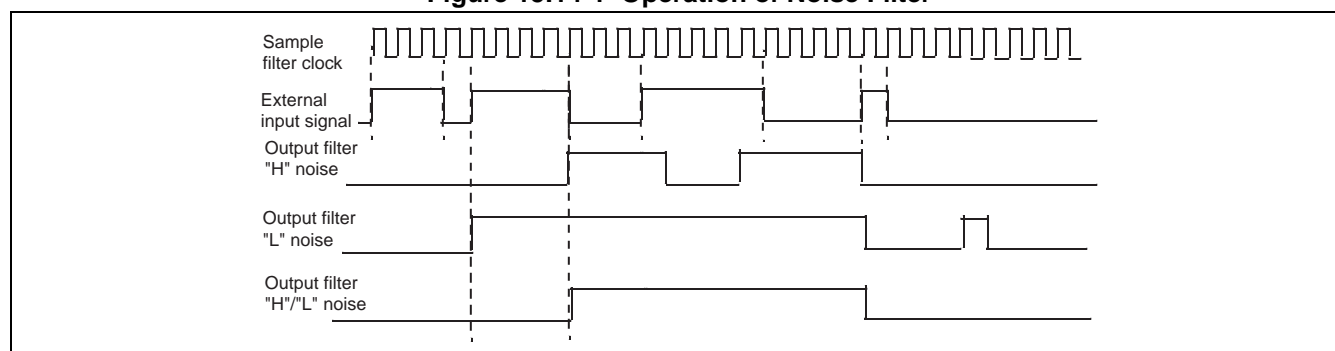


## 15.14 Operating Description of Noise Filter

This section describes the operations of the noise filter for the 8/16-bit compound timer.

When the input capture or PWC timer function has been selected, a noise filter can be used to eliminate the pulse noise of the signal from the external input pin (EC0/EC1). H-pulse noise, L-pulse noise, or H/L-pulse noise elimination can be selected depending on the register setting (TMCr0:FE11, FE10, FE01, FE00). The maximum pulse width from which to eliminate noise is three machine clock cycles. When the filter function is active, the signal input is subject to a delay of four machine clock cycles.

**Figure 15.14-1 Operation of Noise Filter**



## 15.15 States in Each Mode during Operation

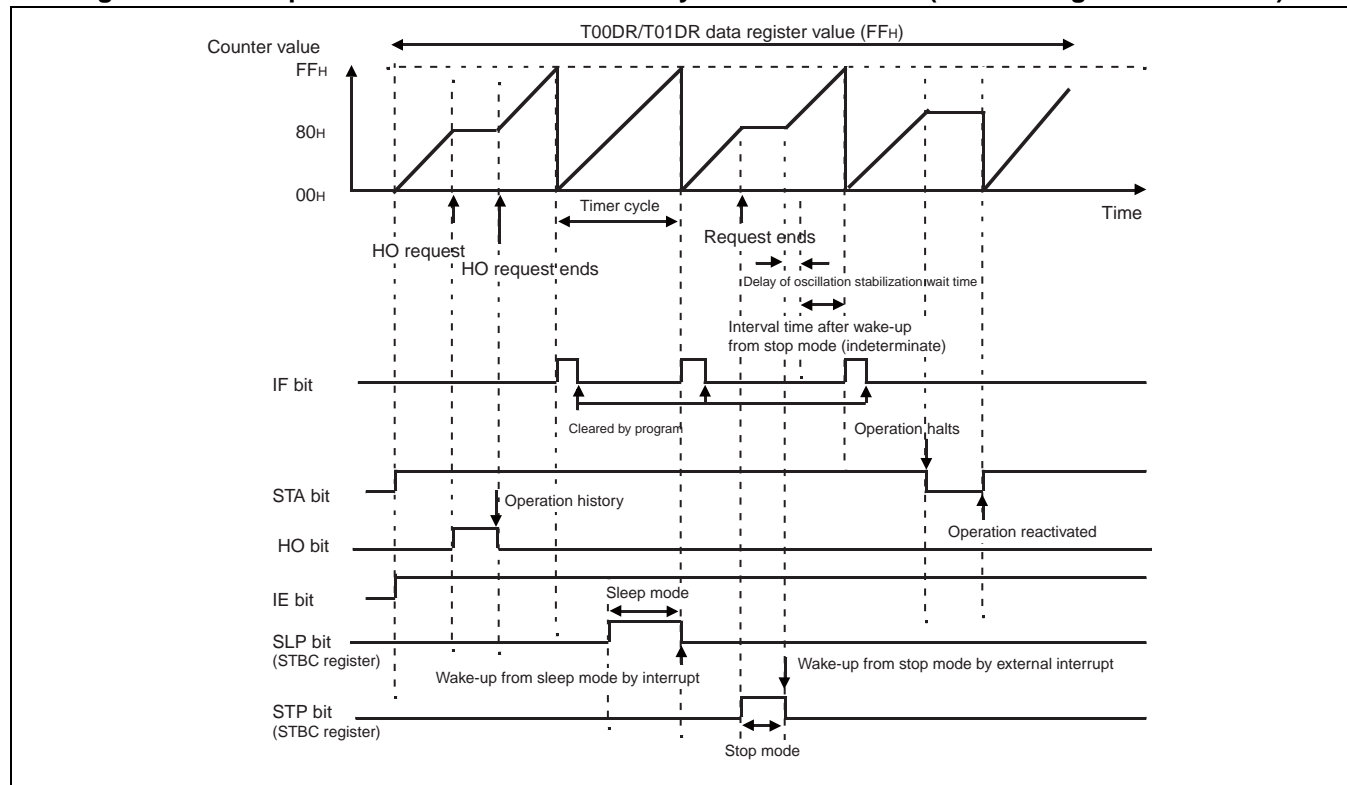
This section describes how the 8/16-bit compound timer behaves when the microcontroller enters watch mode or stop mode or when a suspend (T00CR1/T01CR1:HO = 1) request is issued during operation.

### ■ When Interval Timer, Input Capture, or PWC Function has been Selected

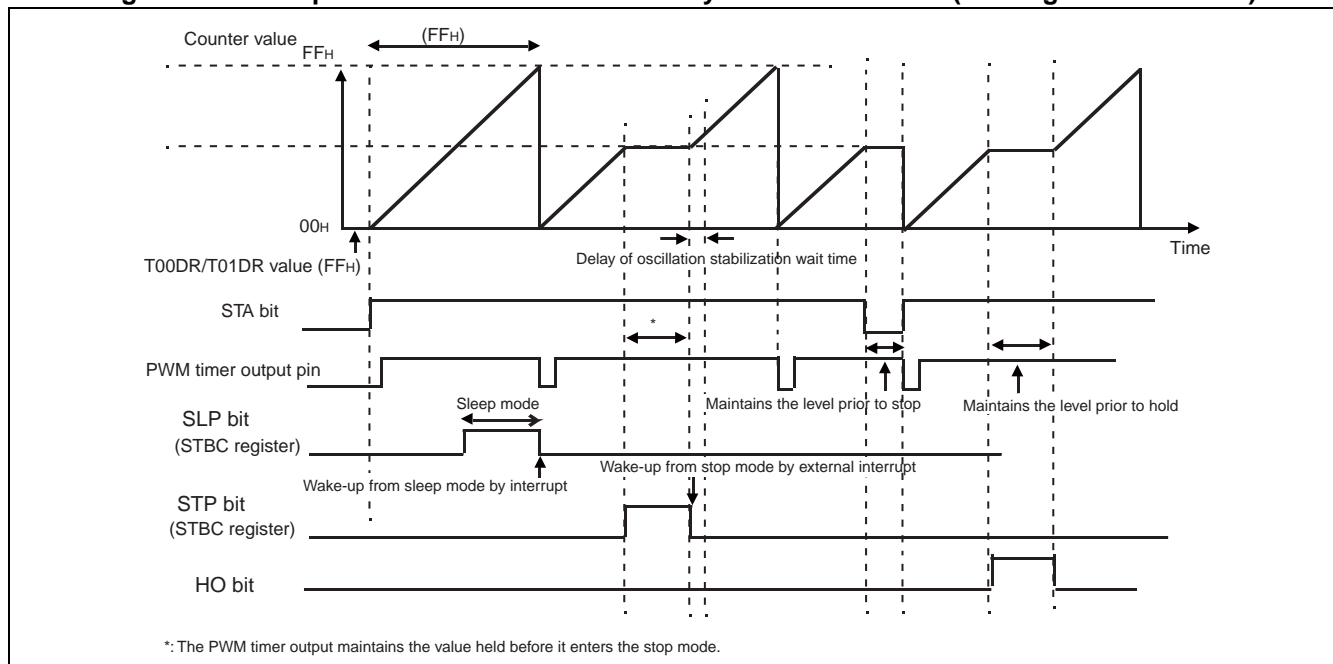
Figure 15.15-1 shows how the counter value changes when transition to watch mode or stop mode or a suspend request occurs during operation of the 8/16-bit compound timer.

The counter stops operation while holding the value when transition to stop mode or watch mode occurs. When the stop mode or watch mode is canceled by an interrupt, the counter resumes operation with the last value held. So the first interval time and external clock count are incorrect. After releasing from stop mode or watch mode, be sure to initialize the counter value.

Figure 15.15-1 Operations of Counter in Standby Mode or in Pause (Not Serving as PWM Timer)



**Figure 15.15-2 Operations of Counter in Standby Mode or in Pause (Serving as PWM Timer)**



## 15.16 Notes on Using 8/16-bit Compound Timer

---

**This section explains the precautions to be taken when using the 8/16-bit compound timer.**

---

### ■ Notes on Using 8/16-bit Compound Timer

When changing the timer function by using the timer operation mode select bits (T00CR0/T01CR0:F3, F2, F1, F0), the timer operation must be stopped (T00CR1/T01CR1:STA = 0) before clearing the interrupt flag (T00CR1/T01CR1:IF, IR), interrupt enable bits (T00CR1/T01CR1:IE, T00CR0/T01CR0:IFE) and buffer full flag (T00CR1/T01CR1:BF).

When the PWC or input capture function has been selected, an interrupt may occur even before the timer is activated (STA = 0). Therefore, nullify the value of the 8/16-bit compound timer 00/01 data register (T00DR/T01DR) obtained before the activation.

In the case of using the input capture function, when the timing at which the 8/16-bit compound timer captures a counter value is the detection of either edge of the external input signal (T00CR0/T01CR0:F3-F0=1100<sub>B</sub> or 1111<sub>B</sub>), the operations in falling edge detection vary according to the level of the external input signal as explained below.

- External input signal level: H

In both free-run mode and clear mode, the first falling edge is ignored, no counter value is transferred to the data register (T00DR/T01DR), and the pulse width measurement completion/edge detection flag (T00CR1/T01CR1:IR) is not set. In addition, in clear mode, the counter is not cleared either.

- External input signal level: L

The 8/16-bit compound timer starts edge detection from the first rising edge.

# ***CHAPTER 16***

---

## ***8/16-BIT PPG***

**This chapter describes the functions and operations of the 8/16-bit PPG.**

- 16.1 Overview of 8/16-bit PPG
- 16.2 Configuration of 8/16-bit PPG
- 16.3 Channels of 8/16-bit PPG
- 16.4 Pins of 8/16-bit PPG
- 16.5 Registers of 8/16-bit PPG
- 16.6 Interrupts of 8/16-bit PPG
- 16.7 Operating Description of 8/16-bit PPG
- 16.8 Notes on Using 8/16-bit PPG
- 16.9 Sample Programs for 8/16-bit PPG Timer

## **16.1 Overview of 8/16-bit PPG**

---

**The 8/16-bit PPG is an 8-bit reload timer module that uses pulse output control based on timer operation to perform PPG output. The 8/16-bit PPG also operates in cascade (8 bits + 8 bits) as 16-bit PPG.**

---

### **■ Overview of 8/16-bit PPG**

The following section summarizes the 8/16-bit PPG functions.

- **8-bit PPG output independent operation mode**

In this mode, the unit can operate as 2 8-bit PPG (PPG timer 00 and PPG timer 01).

- **8-bit prescaler + 8-bit PPG output operation mode**

The rising and falling edge detection pulses from the PPG timer 01 output can be inputted to the down-counter of the PPG timer 00 to enable variable-cycle 8-bit PPG output.

- **16-bit PPG output operation mode**

The unit can also operate in cascade (PPG timer 01 (upper 8 bits) + PPG timer 00 (lower 8 bits)) as 16-bit PPG output.

- **PPG output operation**

In this operation, a variable-cycle pulse waveform is outputted in any duty ratio.

The unit can also be used as a D/A converter in conjunction with an external circuit.

- **Output inversion mode**

This mode can invert the PPG output value.

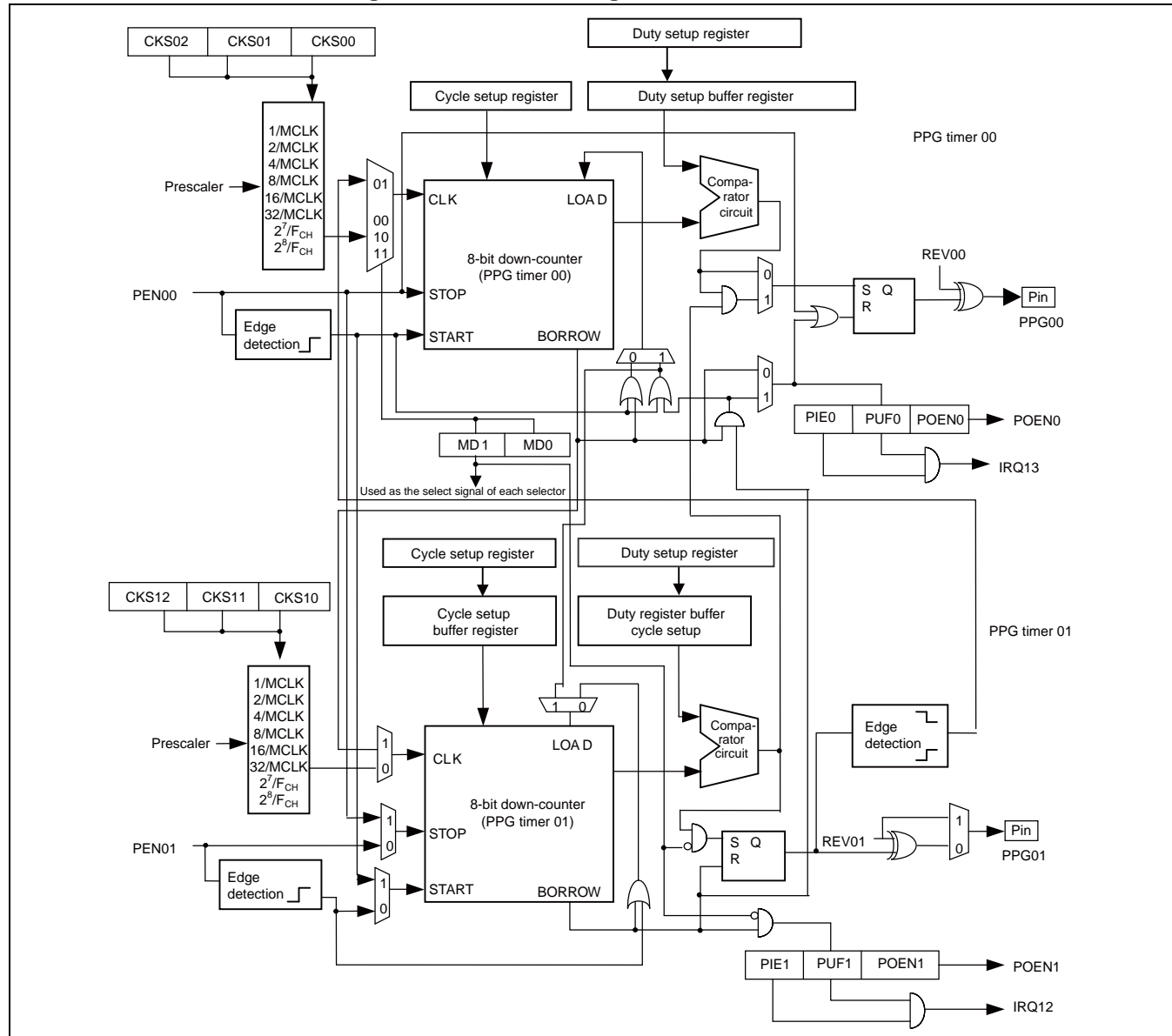
## 16.2 Configuration of 8/16-bit PPG

This section shows the block diagram of 8/16-bit PPG.

### ■ Block Diagram of 8/16-bit PPG

Figure 16.2-1 shows the block diagram of the 8/16-bit PPG.

Figure 16.2-1 Block Diagram of 8/16-bit PPG





● Counter clock selector

The clock for the countdown of 8-bit down counter is selected from eight types of internal count clocks.

● 8-bit down-counter

It counts down with the count clock selected with the count clock selector.

● Comparator circuit

The output is kept "H" level until the value of 8-bit down counter is corresponding to the value of 8/16-bit PPG duty setup buffer register from the value of 8/16-bit set buffer register of PPG cycle.

Afterwards, after keep "L" level the output until the counter value is corresponding to "1", it keeps counting 8-bit down counter from the value of 8/16-bit PPG cycle setup buffer register.

● 8/16-bit PPG timer 01 control register (PC01)

The operation condition on the PPG timer 01 side of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 00 control register (PC00)

The operation mode of 8/16-bit PPG timer and the operation condition on the PPG timer 00 side are set.

● 8/16-bit PPG timer 01/00 cycle setup buffer register ch.0 (PPS01), ch.0(PPS00)

The compare value for the cycle of 8/16-bit PPG timer is set.

● 8/16-bit PPG timer 01/00 duty setup buffer register ch.0 (PDS01), ch.0(PDS00)

The compare value for "H" width of 8/16-bit PPG timer is set.

● 8/16-bit PPG start register

The start or the stop of 8/16-bit PPG timer is set.

● 8/16-bit PPG output inversion register

An initial level also includes the output of 8/16-bit PPG timer and it is reversed.

■ Input Clock

The 8/16-bit PPG uses the output clock from the prescaler as its input clock (count clock).

## 16.3 Channels of 8/16-bit PPG

This section describes the channels of the 8/16-bit PPG.

### ■ Channels of 8/16-bit PPG

MB95110B/M series has two channels of the 8/16-bit PPG. There are 8-bit PPG timer 00 and 8-bit PPG timer 01 in 1 channel. They can be used respectively as two 8-bit PPGs. Also, they can be used as a 16-bit PPG.

Table 16.3-1 and Table 16.3-2 show the channels and their corresponding pins and registers.

**Table 16.3-1 Pins of 8/16-bit PPG**

Channel	Pin name	Pin function
0	PPG00	PPG timer 00 (8-bit PPG (00), 16-bit PPG)
	PPG01	PPG timer 01 (8-bit PPG (01), 8-bit prescaler)
1	PPG10	PPG timer 00 (8-bit PPG (10), 16-bit PPG)
	PPG11	PPG timer 01 (8-bit PPG (11), 8-bit prescaler)

**Table 16.3-2 Registers of 8/16-bit PPG**

Channel	Register name	Corresponding register (as written in this manual)
0	PC01	8/16-bit PPG timer 01 control register
	PC00	8/16-bit PPG timer 00 control register
	PPS01	8/16-bit PPG timer 01 cycle setup buffer register
	PPS00	8/16-bit PPG timer 00 cycle setup buffer register
	PDS01	8/16-bit PPG timer 01 duty setup buffer register
	PDS00	8/16-bit PPG timer 00 duty setup buffer register
1	PC11	8/16-bit PPG timer 01 control register
	PC10	8/16-bit PPG timer 00 control register
	PPS11	8/16-bit PPG timer 01 cycle setup buffer register
	PPS10	8/16-bit PPG timer 00 cycle setup buffer register
	PDS11	8/16-bit PPG timer 01 duty setup buffer register
	PDS10	8/16-bit PPG timer 00 duty setup buffer register
Both channels	PPGS	8/16-bit PPG start register
	REVC	8/16-bit PPG output inversion register

The following sections describe only the 8/16-bit PPG in ch.0 side.

## **16.4 Pins of 8/16-bit PPG**

---

**This section describes the pins of the 8/16-bit PPG.**

---

### **■ Pins of 8/16-bit PPG**

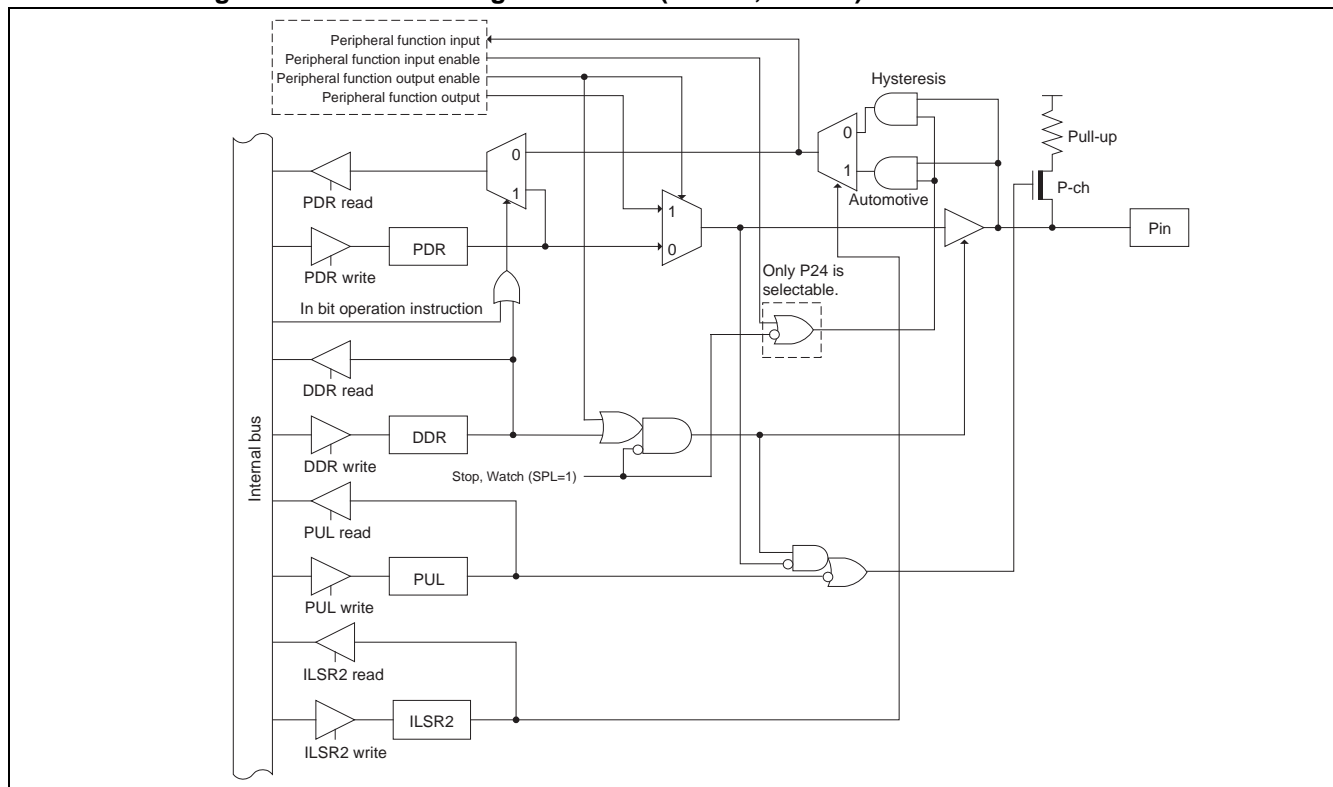
#### **● PPG00 pin and PPG01 pin**

These pins function both as general-purpose I/O ports and 8/16-bit PPG outputs.

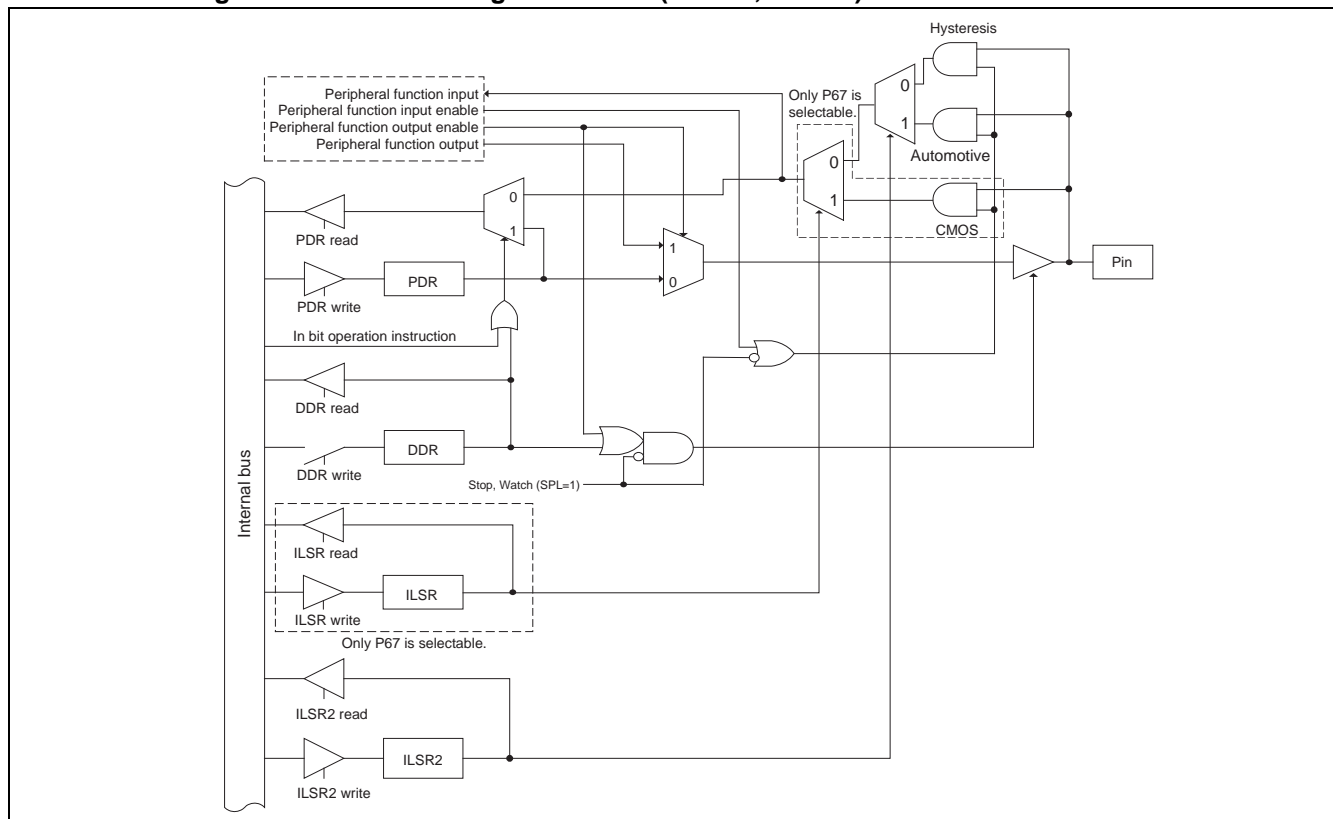
PPG00, PPG01: A PPG waveform is outputted to these pins. The PPG waveform can be outputted by enabling the output by the 8/16-bit PPG timer 01/00 control registers (PC00: POEN0 = 1, PC01: POEN1 = 1).

## ■ Block Diagram of Pins Related to 8/16-bit PPG

**Figure 16.4-1 Block Diagram of Pins (PPG00, PPG01) Related to 8/16-bit PPG**



**Figure 16.4-2 Block Diagram of Pins (PPG10, PPG11) Related to 8/16-bit PPG**



## 16.5 Registers of 8/16-bit PPG

This section describes the registers of the 8/16-bit PPG.

### ■ Registers of 8/16-bit PPG

Figure 16.5-1 shows the registers of the 8/16-bit PPG.

Figure 16.5-1 Registers of 8/16-bit PPG

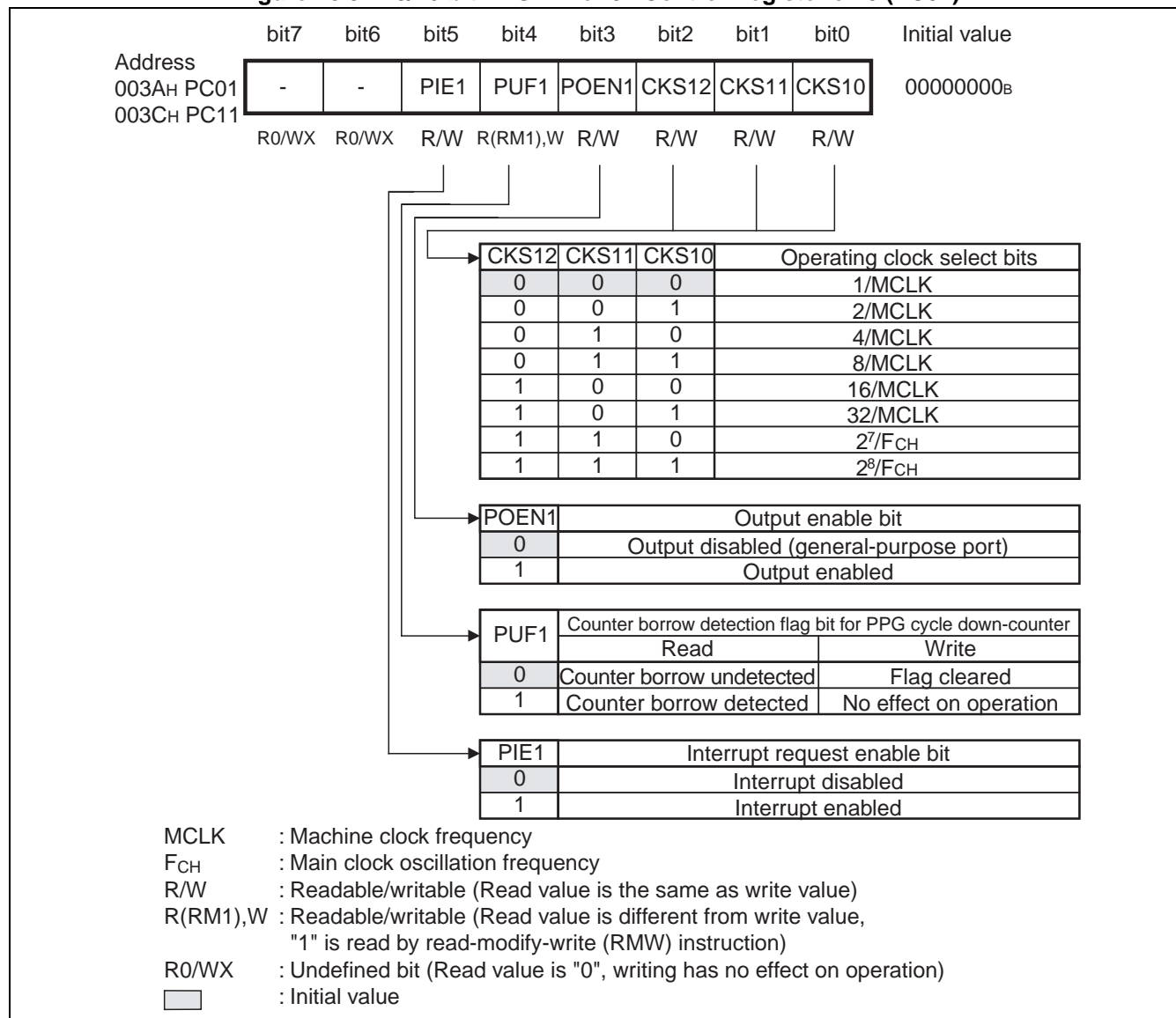
8/16-bit PPG timer 01 control register (PC01)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
003A <sub>H</sub> PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
	R0/WX	R0/WX	R/W	R(RM1),W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
8/16-bit PPG timer 00 control register (PC00)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
003B <sub>H</sub> PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
8/16-bit PPG timer 01 cycle setup buffer register (PPS01)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0F9C <sub>H</sub> PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
11111111 <sub>B</sub>								
8/16-bit PPG timer 00 cycle setup buffer register (PPS00)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0F9D <sub>H</sub> PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
11111111 <sub>B</sub>								
8/16-bit PPG timer 01 duty setup buffer register (PDS01)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0F9E <sub>H</sub> PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
11111111 <sub>B</sub>								
8/16-bit PPG timer 00 duty setup buffer register (PDS00)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0F9F <sub>H</sub> PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
11111111 <sub>B</sub>								
8/16-bit PPG start register (PPGS)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FA4 <sub>H</sub>	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
8/16-bit PPG output inversion register (REVC)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FA5 <sub>H</sub>	-	-	-	-	REV11	REV10	REV01	REV00
	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
R/W: Readable/writable (Read value is the same as write value)								
R(RM1), W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)								
R0/WX: Undefined bit (Read value is "0", writing has no effect on operation)								

## 16.5.1 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)

The 8/16-bit PPG timer 01 control register ch.0 (PC01) sets the operating conditions for PPG timer 01.

## ■ 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)

Figure 16.5-2 8/16-bit PPG Timer 01 Control Register ch.0 (PC01)



**Table 16.5-1 8/16-bit PPG Timer 01 Control Register (PC01)**

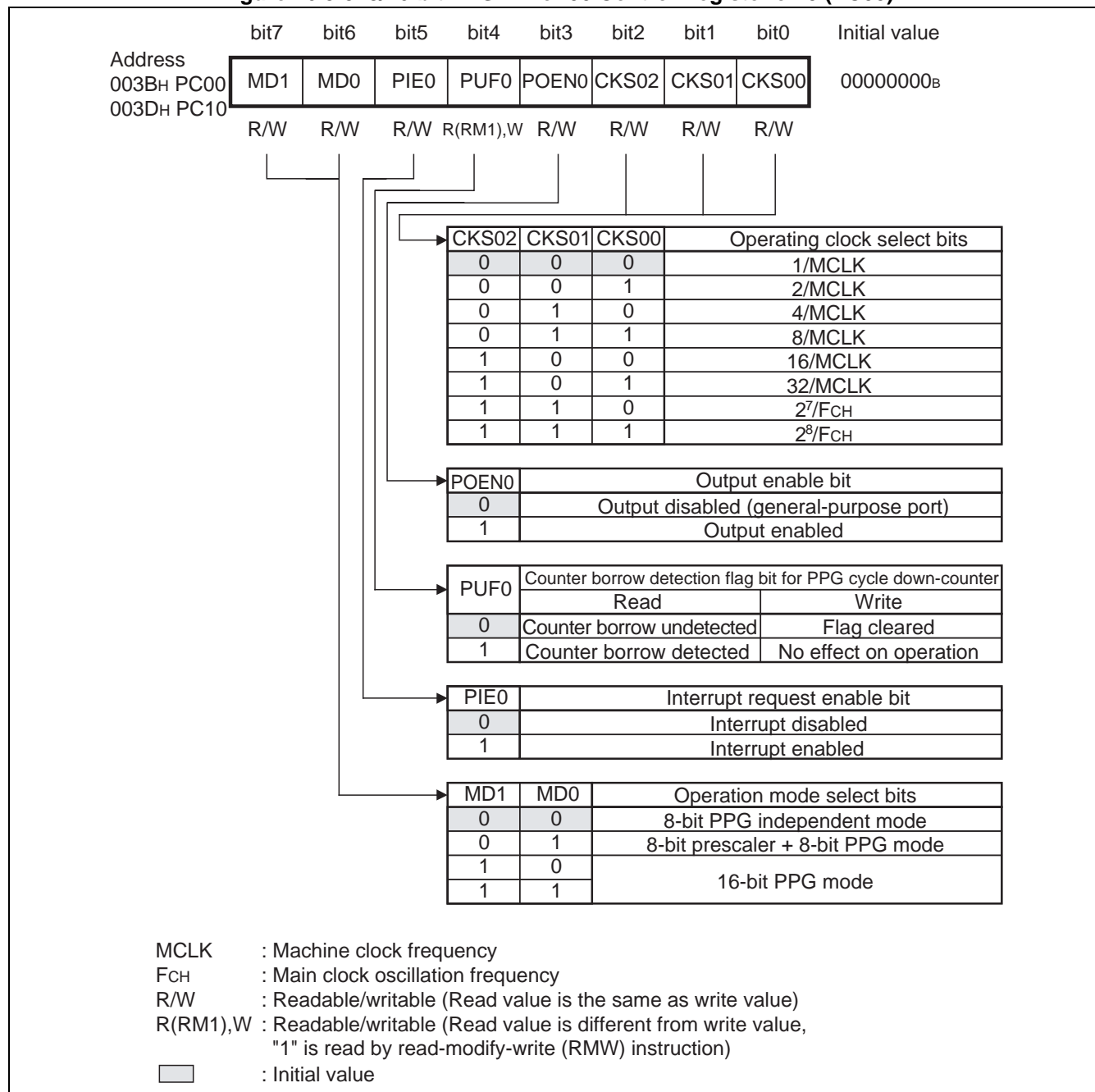
Bit name		Function
bit7, bit6	-: Undefined bits	These bits are undefined. <ul style="list-style-type: none"> <li>• Writing to the bits is meaningless.</li> <li>• Read always returns "0".</li> </ul>
bit5	PIE1: Interrupt request enable bit	This bit controls interrupts of PPG timer 01. <b>Setting the bit to "0"</b> : disables interrupts of PPG timer 01. <b>Setting the bit to "1"</b> : enables interrupts of PPG timer 01. The bit outputs an interrupt request (IRQ) when the counter borrow detection bit (PUF1) and the PIE1 bit are both set to "1".
bit4	PUF1: Counter borrow detection flag bit for PPG cycle down- counter	This bit serves as the counter borrow detection flag for the PPG cycle down-counter of the PPG timer 01. <ul style="list-style-type: none"> <li>• This bit is set to "1" when a counter borrow occurs during 8-bit PPG mode or 8-bit prescaler mode.</li> <li>• In 16-bit PPG mode, this bit is not set to "1" even when a counter borrow occurs.</li> <li>• Writing "1" to the bit is meaningless.</li> <li>• Writing "0" clears the bit.</li> <li>• "1" is read in read-modify-write (RMW) instruction.</li> </ul> <b>When the bit is set to "0"</b> : a counter borrow is undetected. <b>When the bit is set to "1"</b> : a counter borrow is detected.
bit3	POEN1: Output enable bit	This bit enables or disables the output of PPG timer 01 pin. <b>When the bit is set to "0"</b> : the PPG timer 01 pin is used as a general-purpose port. <b>When the bit is set to "1"</b> : the PPG timer 01 pin is used as the PPG output pin. Setting this bit to "1" during 16-bit PPG operation mode sets the PPG timer 01 pin as an output. (The setting value of REV01 is outputted. "L" output is supplied when REV01 is 0.)
bit2, bit1, bit0	CKS12, CKS11, CKS10: Operating clock select bits	These bits select the operating clock for 8-bit down-counter of the PPG timer 01. <ul style="list-style-type: none"> <li>• The operating clock is generated from the prescaler. Refer to "CHAPTER 6 CLOCK CONTROLLER".</li> <li>• In 16-bit PPG operation mode, the setting of this bit has no effect on the operation.</li> </ul> <b>"000<sub>B</sub>"</b> : 1/MCLK <b>"001<sub>B</sub>"</b> : 2/MCLK <b>"010<sub>B</sub>"</b> : 4/MCLK <b>"011<sub>B</sub>"</b> : 8/MCLK <b>"100<sub>B</sub>"</b> : 16/MCLK <b>"101<sub>B</sub>"</b> : 32/MCLK <b>"110<sub>B</sub>"</b> : $2^7/F_{CH}$ <b>"111<sub>B</sub>"</b> : $2^8/F_{CH}$ Note: Use of a sub clock (in dual clock product) stops the time-base timer operation. Therefore, selecting "110 <sub>B</sub> " or "111 <sub>B</sub> " is prohibited.

## 16.5.2 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)

The 8/16-bit PPG timer 00 control register ch.0 (PC00) sets the operating conditions and the operation mode for PPG timer 00.

## ■ 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)

Figure 16.5-3 8/16-bit PPG Timer 00 Control Register ch.0 (PC00)





**Table 16.5-2 8/16-bit PPG0 Control Register (PC0)**

Bit name		Function
bit7, bit6	MD1, MD0: Operation mode select bits	These bits select the PPG operation mode. Do not modify the bit settings during counting. <b>When set to "00<sub>B</sub>"</b> : 8-bit PPG independent mode <b>When set to "01<sub>B</sub>"</b> : 8-bit prescaler + 8-bit PPG mode <b>When set to "1x<sub>B</sub>"</b> : 16-bit PPG mode
bit5	PIE0: Interrupt request enable bit	This bit controls interrupts of PPG timer 00. <ul style="list-style-type: none"> <li>Set this bit in 16-bit PPG operation mode.</li> </ul> <b>Setting the bit to "0"</b> : disables interrupts of PPG timer 00. <b>Setting the bit to "1"</b> : enables interrupts of PPG timer 00. <ul style="list-style-type: none"> <li>An interrupt request (IRQ) is outputted when the counter borrow detection bit (PUF0) and PIE0 bit are both set to "1".</li> </ul>
bit4	PUF0: Counter borrow detection flag bit for PPG cycle down-counter	This is the counter borrow detection flag for the PPG cycle down-counter of PPG timer 00. <ul style="list-style-type: none"> <li>Only this bit is effective in 16-bit PPG operation mode (PC01:PUF1 is not operable). Note: Always effective in 8-bit mode</li> <li>Writing "1" to this bit is meaningless.</li> <li>Writing "0" clears the bit.</li> <li>"1" is read in read-modify-write (RMW) instruction.</li> </ul> <b>When set to "0"</b> : Counter borrow of PPG timer 00 undetected <b>When set to "1"</b> : Counter borrow of PPG timer 00 detected
bit3	POEN0: Output enable bit	This bit enables or disables the output of PPG timer 00 pin. <b>When set to "0"</b> : PPG timer 00 pin is used as a general-purpose port. <b>When set to "1"</b> : PPG timer 00 pin is used as the PPG output pin. As the output is supplied from the PPG timer 00 pin in 16-bit PPG operation mode, this bit is used to control the operation.
bit2, bit1, bit0	CKS02, CKS01, CKS00: Operating clock select bits	These bits select the operating clock for PPG down-counter PPG timer 00. <ul style="list-style-type: none"> <li>The operating clock is generated from the prescaler. Refer to "CHAPTER 6 CLOCK CONTROLLER".</li> <li>The rising and falling edge detection pulses from the PPG timer 01 output are used as the count clock for PPG timer 00 when the 8-bit prescaler + 8-bit PPG mode has been selected. Therefore, the setting of this bit has no effect on the operation.</li> <li>Set this bit in 16-bit PPG operation mode.</li> </ul> <b>"000<sub>B</sub>"</b> : 1/MCLK <b>"001<sub>B</sub>"</b> : 2/MCLK <b>"010<sub>B</sub>"</b> : 4/MCLK <b>"011<sub>B</sub>"</b> : 8/MCLK <b>"100<sub>B</sub>"</b> : 16/MCLK <b>"101<sub>B</sub>"</b> : 32/MCLK <b>"110<sub>B</sub>"</b> : $2^7/F_{CH}$ <b>"111<sub>B</sub>"</b> : $2^8/F_{CH}$ Note: Use of a sub clock (in dual clock product) stops the time-base timer operation. Therefore, selecting "110 <sub>B</sub> " or "111 <sub>B</sub> " is prohibited.

### 16.5.3 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

The 8/16-bit PPG timer 00/01 cycle setup buffer register (PPS01), (PPS00) sets the PPG output cycle.

#### ■ 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

Figure 16.5-4 8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00)

PPS01	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	11111111 <sub>B</sub>
0F9C <sub>H</sub> PPS01	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0FA0 <sub>H</sub> PPS11									
PPS00	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	11111111 <sub>B</sub>
0F9D <sub>H</sub> PPS00	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0FA1 <sub>H</sub> PPS10									

R/W : Readable/writable (Read value is the same as write value)

This register is used to set the PPG output cycle.

- In 16-bit PPG mode, PPS01 serves as the upper 8 bits, while PPS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load.
- 8-bit mode: Cycle = max. 255 (FF<sub>H</sub>) × Input clock cycle
- 16-bit mode: Cycle = max. 65535 (FFFF<sub>H</sub>) × Input clock cycle
- Initialized at reset.
- Do not set the cycle to "00<sub>H</sub>" or "01<sub>H</sub>" when using the unit in 8-bit PPG independent mode, or in 8-bit prescaler mode + 8-bit PPG mode
- Do not set the cycle to "0000<sub>H</sub>" or "0001<sub>H</sub>" when using the unit in 16-bit PPG mode.
- If the cycle settings are modified during the operation, the modified settings will be effective from the next PPG cycle.

## 16.5.4 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

The 8/16-bit PPG timer 00/01 duty setup buffer register (PDS01), (PDS00) sets the duty of the PPG output.

### ■ 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)

**Figure 16.5-5 8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00)**

PDS01	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0	11111111 <sub>B</sub>
0F9E <sub>H</sub> PDS01	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0FA2 <sub>H</sub> PDS11									
PDS00	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0	11111111 <sub>B</sub>
0F9F <sub>H</sub> PDS00	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0FA3 <sub>H</sub> PDS10									

R/W : Readable/writable (Read value is the same as write value)

This register is used to set the duty of the PPG output ("H" pulse width when normal polarity).

- In 16-bit PPG mode, PDS01 serves as the upper 8 bits while PDS00 serves as the lower 8 bits.
- In 16-bit PPG mode, write the upper bits before the lower bits. When only the upper bits are written, the previously written value is reused in the next load. By writing to PDS00, PDS01 is updated.
- Initialized at reset.
- To set the duty to 0%, select "00<sub>H</sub>".
- To set the duty to 100%, set it to the same value as the 8/16-bit PPG timer 00/01 cycle setup register (PPS00, 01).
- When the 8/16-bit PPG timer 00/01 duty setup register (PDS) is set to a larger value than the setting value of the 8/16-bit PPG cycle setup buffer register (PPS), the PPG output becomes "L" output in the normal polarity (when the output level inversion bit of 8/16-bit PPG output inversion register is "0").
- If the duty settings are modified during operation, the modified value will be effective from the next PPG cycle.

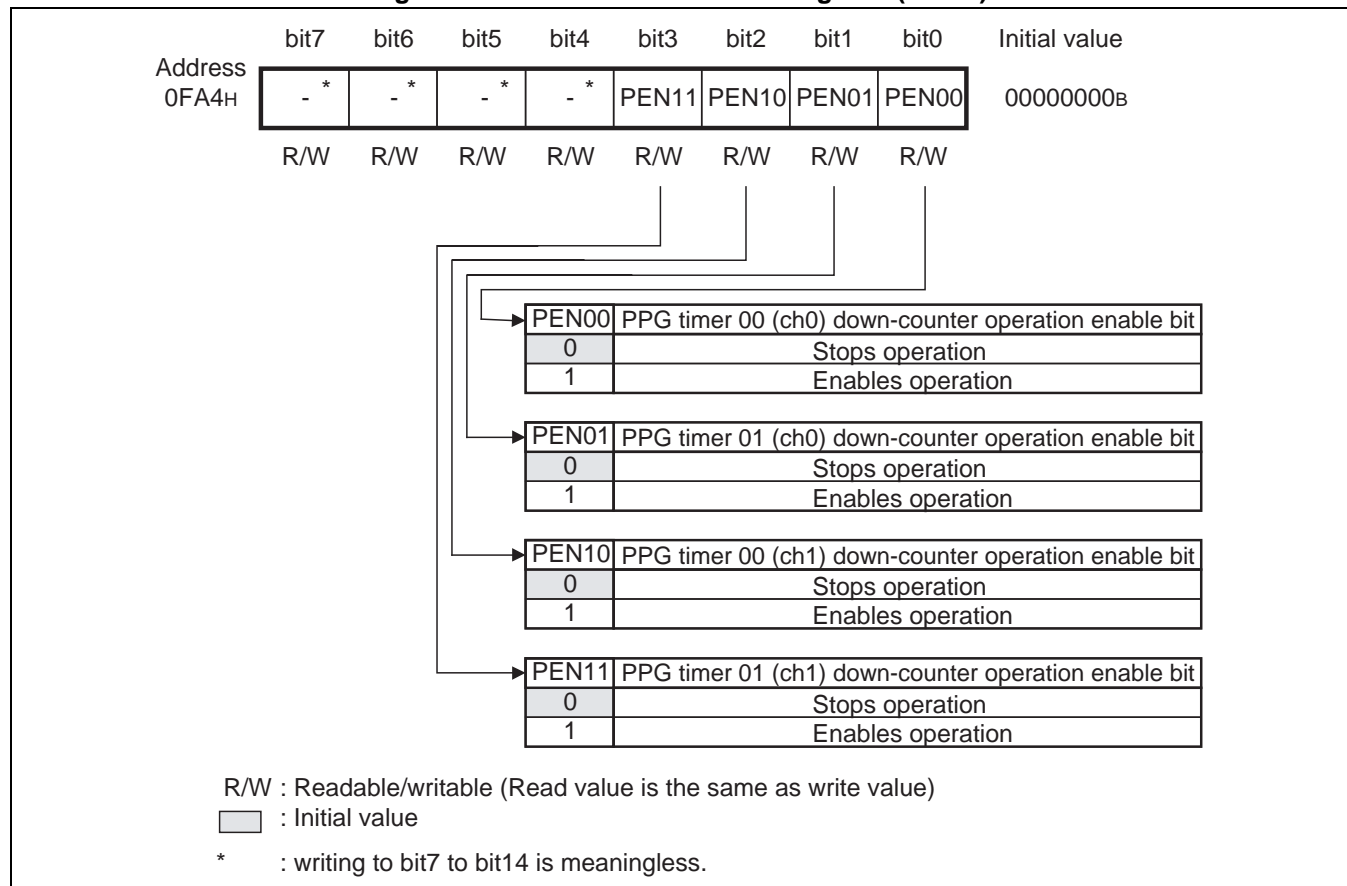
## MB95110B/M Series

### 16.5.5 8/16-bit PPG Start Register (PPGS)

The 8/16-bit PPG start register (PPGS) starts or stops the down-counter. The operation enable bit of each channel is assigned to the PPGS register, allowing simultaneous activation of the PPG channels.

#### ■ 8/16-bit PPG Start Register (PPGS)

Figure 16.5-6 8/16-bit PPG Start Register (PPGS)

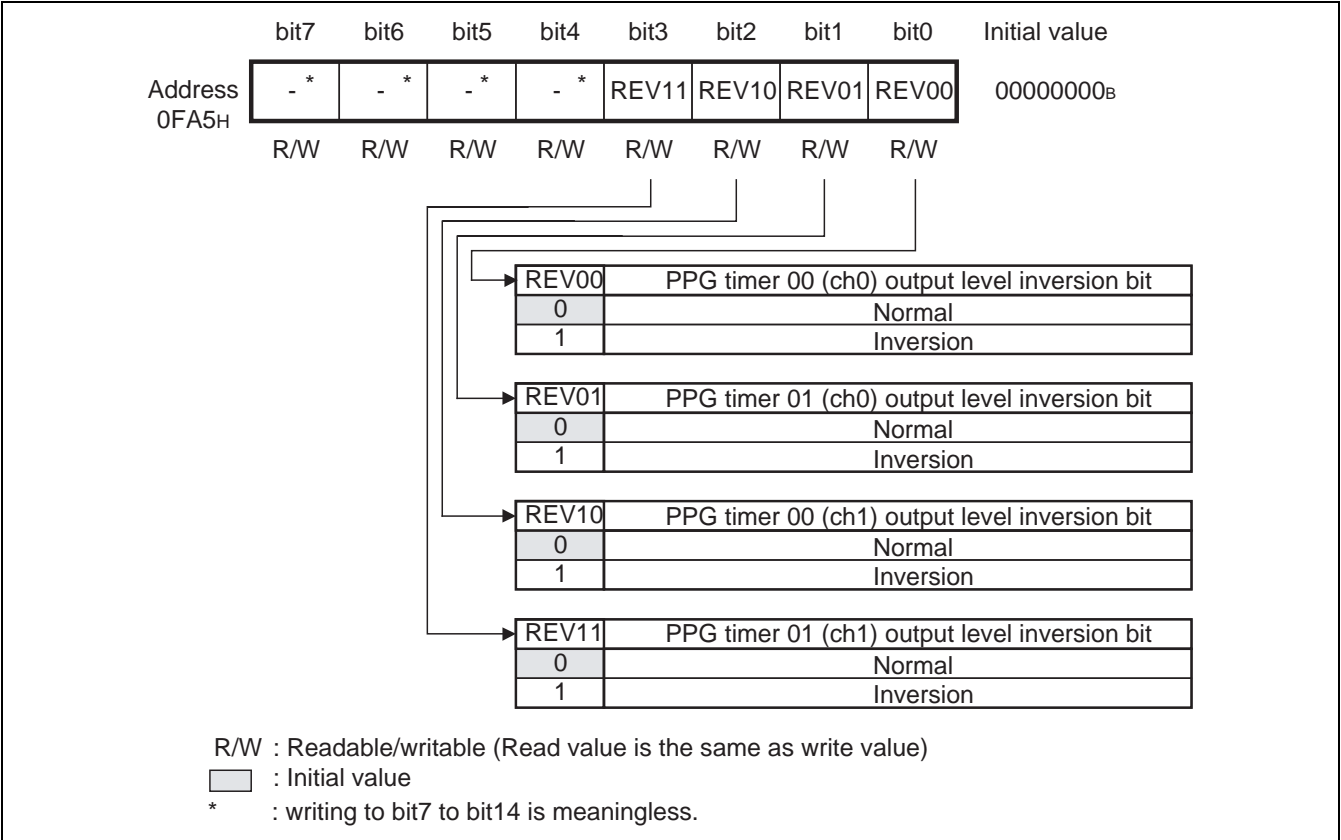


16.5.6 8/16-bit PPG Output Inversion Register (REVC)

The 8/16-bit PPG output inversion register (REVC) inverts the PPG output including the initial level.

■ 8/16-bit PPG Output Inversion Register (REVC)

Figure 16.5-7 8/16-bit PPG Output Inversion Register (REVC)



## 16.6 Interrupts of 8/16-bit PPG

The 8/16-bit PPG outputs an interrupt request when a counter borrow is detected.

### ■ Interrupts of 8/16-bit PPG

Table 16.6-1 shows the interrupt control bits and interrupt sources of the 8/16-bit PPG.

**Table 16.6-1 Interrupt Control Bits and Interrupt Sources of 8/16-bit PPG**

Item	Description	
	PPG timer 01 (8-bit PPG, 8-bit prescaler)	PPG timer 00 (8-bit PPG, 16-bit PPG)
Interrupt request flag bit	PUF1 bit in PC01	PUF0 bit in PC00
Interrupt request enable bit	PIE1 bit in PC01	PIE0 bit in PC00
Interrupt source	Counter borrow of PPG cycle down-counter	

When a counter borrow occurs on the down-counter, the 8/16-bit PPG sets the counter borrow detection flag bit (PUF) in the 8/16-bit PPG timer 00/01 control register (PC) to "1". When the interrupt request enable bit is enabled (PIE = 1), an interrupt request is outputted to the interrupt controller.

In 16-bit PPG mode, the 8/16-bit PPG timer 00 control register (PC00) is available.

### ■ Registers and Vector Table Related to Interrupts of 8/16-bit PPG

**Table 16.6-2 Registers and Vector Table Related to Interrupts of 8/16-bit PPG**

Interrupt source	Interrupt request No.	Interrupt level setup register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.1 (lower)*	IRQ9	ILR2	L09	FFE8 <sub>H</sub>	FFE9 <sub>H</sub>
ch.1 (upper)	IRQ10	ILR2	L10	FFE6 <sub>H</sub>	FFE7 <sub>H</sub>
ch.0 (upper)*	IRQ12	ILR3	L12	FFE2 <sub>H</sub>	FFE3 <sub>H</sub>
ch.0 (lower)	IRQ13	ILR3	L13	FFE0 <sub>H</sub>	FFE1 <sub>H</sub>

\*: ch.1 (lower) shares the same interrupt request number and vector table as UART/SIO (ch.1) and ch.0 (upper) shares them as I<sup>2</sup>C (ch.1).

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

## **16.7 Operating Description of 8/16-bit PPG**

---

**This section describes the operations of the 8/16-bit PPG.**

---

### **■ Setup Procedure Example**

The setup procedure of the 8/16-bit PPG is described below.

#### **● Initial setting**

- 1) Set the port output (DDR2, DDR6)
- 2) Set the interrupt level (ILR2, ILR3)
- 3) Select the operating clock, enable the output and interrupt (PC01)
- 4) Select the operating clock, enable the output and interrupt, select the operation mode (PC00)
- 5) Set the cycle (PPS)
- 6) Set the duty (PDS)
- 7) Set the output inversion (REVC)
- 8) Start PPG (PPGS)

#### **● Interrupt processing**

- 1) Process any interrupt
- 2) Clear the interrupt request flag (PC01: PUF1, PC00: PUF0)
- 3) Start PPG (PPGS)

## 16.7.1 8-bit PPG Independent Mode

In this mode, the unit operates as two channels (PPG timer 00 and PPG timer 01) of the 8-bit PPG.

### ■ Setting 8-bit Independent Mode

The unit requires the register settings shown in Figure 16.7-1 to operate in 8-bit independent mode.

Figure 16.7-1 8-bit Independent Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			⊙	⊙	⊙	⊙	⊙	⊙
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0	⊙	⊙	⊙	⊙	⊙	⊙
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	⊙	⊙
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	⊙	⊙

⊙ : Used bit  
 0 : Set "0"  
 \* : The bit status depends on the number of channels provided.

### ■ Operation of 8-bit PPG Independent Mode

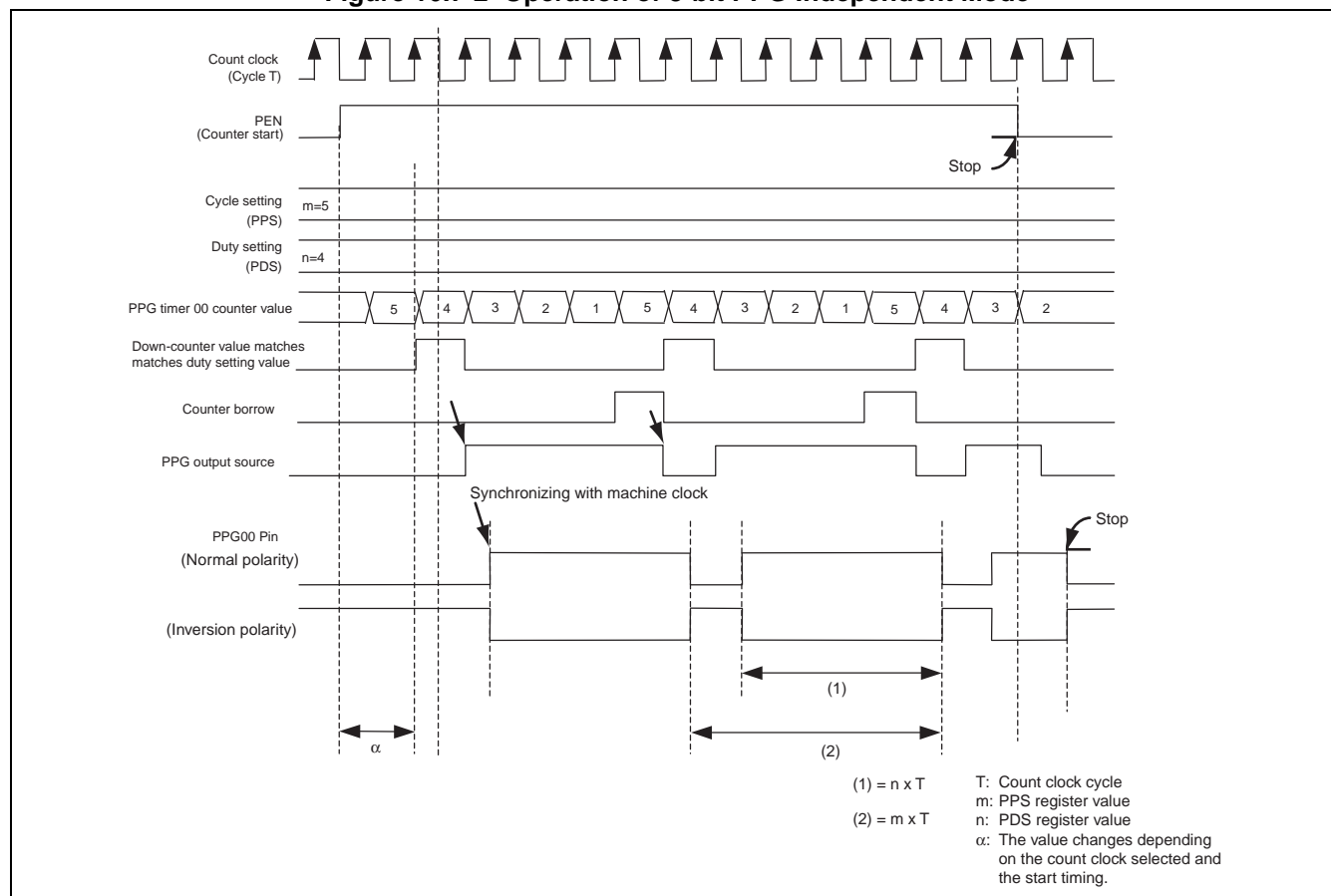
- This mode is selected when the operation mode select bits (MD1, 0) in the 8/16-bit PPG timer 00 control register (PC00) are set to "00<sub>B</sub>".
- When the corresponding bit (PEN) in the 8/16-bit PPG start register (PPGS) is set to "1", the value in the 8/16-bit PPG cycle setup buffer register (PPS) is loaded to start down-count operation. When the count value reaches "1", the value in the cycle setup register is reloaded to repeat the counting.
- "H" is output to the PPG output synchronizing with the count clock. When the down-counter value matches the value in the 8/16-bit PPG timer 00/01 duty setup buffer register (PDS). After "H" which is the value of duty setting is output, "L" is output to the PPG output.

If, however, the PPG output inversion bit is set to "1", the PPG output is set and reset inversely from the above process.

Figure 16.7-2 shows the operation of the 8-bit PPG independent mode.



**Figure 16.7-2 Operation of 8-bit PPG Independent Mode**



Example for setting the duty to 50%

When PDS is set to "02<sub>H</sub>" with PPS set to "04<sub>H</sub>", the PPG output is set at a duty ratio of 50% (PPS setting value / 2 set to PDS).

## 16.7.2 8-bit Prescaler + 8-bit PPG Mode

In this mode, the rising and falling edge detection pulses from the PPG timer 01 output can be used as the count clock of the PPG timer 00 down-counter to allow variable-cycle 8-bit PPG output from PPG timer 00.

### ■ Setting 8-bit Prescaler + 8-bit PPG Mode

The unit requires the register settings shown in Figure 16.7-3 to operate in 8-bit prescaler + 8-bit PPG mode.

Figure 16.7-3 Setting 8-bit Prescaler + 8-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			⊙	⊙	⊙	⊙	⊙	⊙
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	1	⊙	⊙	⊙	x	x	x
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	⊙	⊙
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	⊙	⊙

⊙ : Used bit

0 : Set "0"

1 : Set "1"

x : Setting nullified

\* : The bit status varies depending of the number of channels implemented

### ■ Operation of 8-bit Prescaler + 8-bit PPG Mode

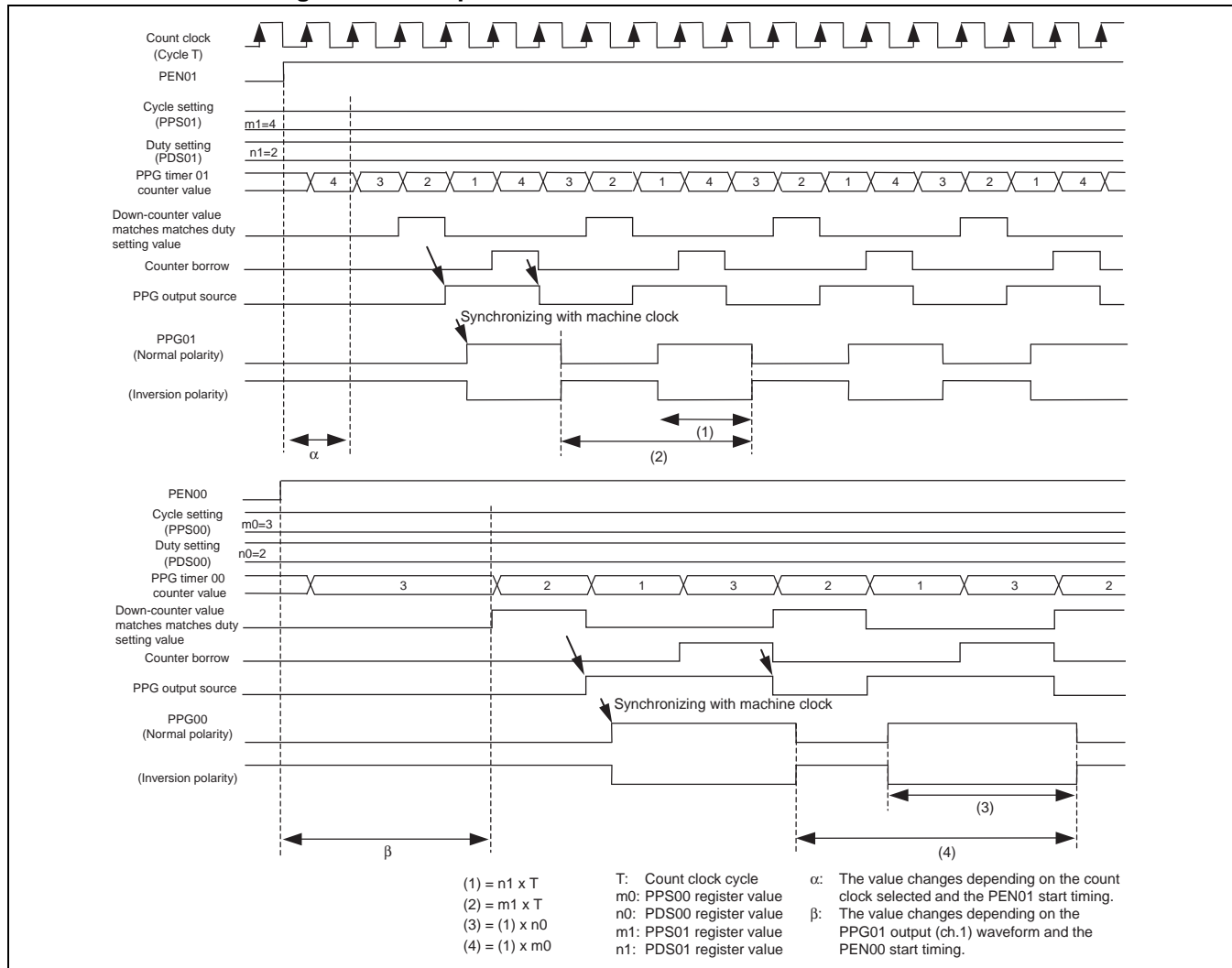
- This mode is selected by setting the operation mode select bits (MD1, 0) of the 8/16-bit PPG timer 00 control register (PC00) to "01<sub>B</sub>". This allows PPG timer 01 to be used as an 8-bit prescaler and PPG timer 00 to be used as an 8-bit PPG.
- When the PPG timer 01 (ch.0) down counter operation enable bit (PEN01) is set to "1", the 8-bit prescaler (PPG timer 01) loads the value in the 8/16-bit PPG timer 01 cycle setup buffer register (PPS01) and starts down-count operation. When the value of the down-counter matches the value in the 8/16-bit PPG timer 01 duty setup buffer register (PDS01), the PPG01 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG01 output is set to "L". If the output inversion signal (REV01) is "0", the polarity will remain the same. If it is "1", the

polarity will be inverted and the signal will be outputted to the PPG pin.

- When the PPG operation enable bit (PEN00) is set to "1", the 8-bit PPG (PPG timer 00) loads the value in the 8/16-bit PPG timer 00 cycle setup buffer register (PPS00) and starts down-count operation (count clock = rising and falling edge detection pulses of PPG01 output after PPG timer 01 operation is enabled). When the count value reaches "1", the value in the 8/16-bit PPG timer 00 cycle setup buffer register is reloaded to repeat the counting. When the value of the down-counter matches the value in the 8/16-bit PPG timer 00 duty setup buffer register (PDS00), the PPG00 output is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 output is reset to "L". If the output inversion signal (REV00) is "0", the polarity will remain the same. If it is "1", the polarity will be inverted and the signal will be outputted to the PPG00 pin.
- Set that the duty of the 8-bit prescaler (PPG timer 01) output to 50%.
- When PPG timer 00 is started with the 8-bit prescaler (PPG timer 01) being stopped, PPG timer 00 does not count.
- When the duty of the 8-bit prescaler (PPG timer 01) is set to 0% or 100%, PPG timer 00 does not perform counting as the 8-bit prescaler (PPG timer 01) output does not toggle.

Figure 16.7-4 shows the operation of 8-bit prescaler + 8-bit PPG mode.

Figure 16.7-4 Operation of 8-bit Prescaler + 8-bit PPG Mode



### 16.7.3 16-bit PPG Mode

In this mode, the unit can operate as a 16-bit PPG when PPG timer 01 and PPG timer 00 are assigned to the upper and lower bits respectively.

#### ■ Setting 16-bit PPG Mode

The unit requires the register settings shown in Figure 16.7-5 to operate in 16-bit PPG mode.

Figure 16.7-5 Setting 16-bit PPG Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
PC01	-	-	PIE1	PUF1	POEN1	CKS12	CKS11	CKS10
			⊙	⊙	⊙	⊙	⊙	⊙
PC00	MD1	MD0	PIE0	PUF0	POEN0	CKS02	CKS01	CKS00
	0	0/1	⊙	⊙	⊙	⊙	⊙	⊙
PPS01	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
	Set PPG output cycle (Upper 8 bits) for PPG timer 01							
PPS00	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
	Set PPG output cycle (Lower 8 bits) for PPG timer 00							
PDS01	DH7	DH6	DH5	DH4	DH3	DH2	DH1	DH0
	Set PPG output duty (Upper 8 bits) for PPG timer 01							
PDS00	DL7	DL6	DL5	DL4	DL3	DL2	DL1	DL0
	Set PPG output duty (Lower 8 bits) for PPG timer 00							
PPGS	-	-	-	-	PEN11	PEN10	PEN01	PEN00
	*	*	*	*	*	*	x	⊙
REVC	-	-	-	-	REV11	REV10	REV01	REV00
	*	*	*	*	*	*	x	⊙

⊙ : Used bit

0 : Set "0"

1 : Set "1"

x : Setting nullified

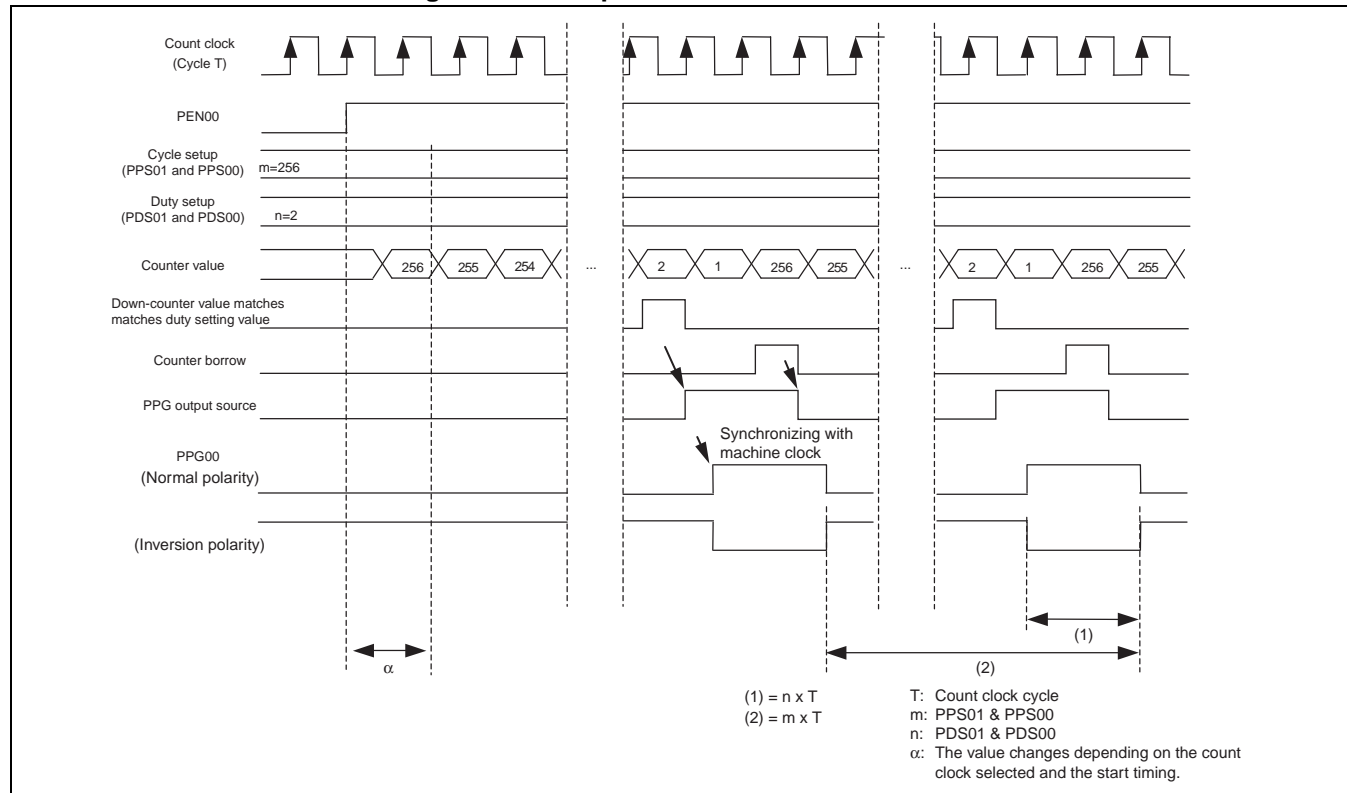
\* : The bit status changes depending on the number of channels implemented.

## ■ Operation of 16-bit PPG Mode

- This mode is selected by setting the operation mode select bits (MD1, 0) of the PPG timer 00 control register (PC00) to "10<sub>B</sub>" or "11<sub>B</sub>".
- When the PPG operation enable bit (PEN00) is set to "1" in 16-bit PPG mode, the 8-bit down-counters (PPG timer 00) and 8-bit down-counter (PPG timer 01) load the values in the 8/16-bit PPG timer 00/01 cycle setup buffer registers (PPS01 for PPG timer 01 and PPS00 for PPG timer 00) and start down-count operation. When the count value reaches "1", the values in the cycle setup register are reloaded and the counters repeat the counting.
- When the values of the down-counters match the values in the 8/16-bit PPG timer duty setup buffer registers (both the value in PDS01 for PPG timer 01 and the value in PDS00 for PPG timer 00), the PPG00 pin is set to "H" synchronizing with the count clock. After "H" which is the value of duty setting is output, the PPG00 pin is set to "L". If the output inversion signal (REV00) is "0", the signal will be outputted to the PPG00 with the polarity unchanged. If it is set to "1", the polarity will be inverted and the signal will be outputted to the PPG00 pin. (ch.0 only. ch.1 will be set to the initial value <"L" if REV01 is "0", or "H" if it is "1">.)

Figure 16.7-6 shows the operation of 16-bit PPG mode.

Figure 16.7-6 Operation of 16-bit PPG Mode



## 16.8 Notes on Using 8/16-bit PPG

---

**The following precautions must be followed when using the 8/16-bit PPG.**

---

### ■ Notes on Using 8/16-bit PPG

#### ● Operational precaution

Depending on the timing between the activation of PPG and count clock, an error may occur in the first cycle of the PPG output immediately after the activation. The error varies depending on the count clock selected. The output, however, is performed properly in the succeeding cycles.

#### ● Precaution regarding interrupts

A PPG interrupt is generated when the interrupt enable bit (PIE1/PIE0) is set to "1" and the interrupt request flag bit (PUF1/PUF0) in the 8/16-bit PPG timer 01/00 control register (PC01/PC00) is also set to "1". Always clear the interrupt request flag bit (PUF1/PUF0) to "0" in the interrupt routine.

## 16.9 Sample Programs for 8/16-bit PPG Timer

We provide sample programs that can be used to operate the 8/16-bit PPG timer.

### ■ Sample Programs for 8/16-bit PPG Timer

For information about the sample programs for the 8/16-bit PPG timer, refer to "■ Sample Programs" in Preface.

### ■ Setup Methods without Sample Program

#### ● How to enable/stop PPG operation

The PPG operation enable bit (PPGS: PEN00 or PEN10) is used for PPG00.

Control	PPG operation enable bit (PEN00 or PEN10)
When stopping PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

PPG operation must be enabled before the PPG is activated.

The PPG operation enable bit (PPGS: PEN01 or PEN11) is used for PPG timer 01..

Control	PPG operation enable bit (PEN01 or PEN11)
When stopping PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

#### ● How to set the PPG operation mode

The operation mode select bits (PC00.MD[1:0]) are used.

#### ● How to select the operating clock

ch.1 is selected by the operating clock select bits (PC01.CKS12/CKS11/CKS10).

ch.0 is selected by the operating clock select bits (PC00.CKS02/CKS01/CKS00).

#### ● How to enable/disable the PPG output pin

The output enable bit (PC00 or PC01.POEN0 or POEN1) is used.

Control	Output enable bit (POEN0 or POEN1)
When enabling PPG output	Set the bit to "1"
When disabling PPG output	Set the bit to "0"

● How to invert the PPG output

The output level inversion bit (REVC.REV00 or REV10) is used for PPG00.

Control	Output level inversion bit (REV00 or REV10)
When inverting PPG output	Set the bit to "1"

The output level inversion bit (REVC.REV01 or REV11) is used for PPG01.

Control	Output level inversion bit (REV01 or REV11)
When inverting PPG output	Set the bit to "1"

● Interrupt-related register

The interrupt level is set by the interrupt setup register shown in the following table.

Interrupt source	Interrupt level setup register	Interrupt vector
ch.1 (lower)	Interrupt level register (ILR2) Address:0007B <sub>H</sub>	#09 Address:0FFE8 <sub>H</sub>
ch.1 (upper)	Interrupt level register (ILR2) Address:0007B <sub>H</sub>	#10 Address:0FFE6 <sub>H</sub>
ch.0 (lower)	Interrupt level register (ILR3) Address:0007C <sub>H</sub>	#12 Address:0FFE2 <sub>H</sub>
ch.0 (upper)	Interrupt level register (ILR3) Address:0007C <sub>H</sub>	#13 Address:0FFE0 <sub>H</sub>

● How to enable/disable/clear interrupts

Interrupt request enable flag, Interrupt request flag

The interrupt request enable bit (PC00 or PC01.PIE0 or PIE1) is used to enable or disable interrupts.

Operation	Interrupt request enable bit (PIE0 or PIE1)
When disabling interrupt requests	Set the bit to "0"
When enabling interrupt requests	Set the bit to "1"

The interrupt request flag (PC00 or PC01.PUF0 or PUF1) is used to clear interrupt requests.

Operation	Interrupt request flag (PUF0 or PUF1)
When clearing interrupt requests	Write "0"





# **CHAPTER 17**

---

## ***16-BIT PPG TIMER***

**This chapter describes the functions and operations of the 16-bit PPG timer.**

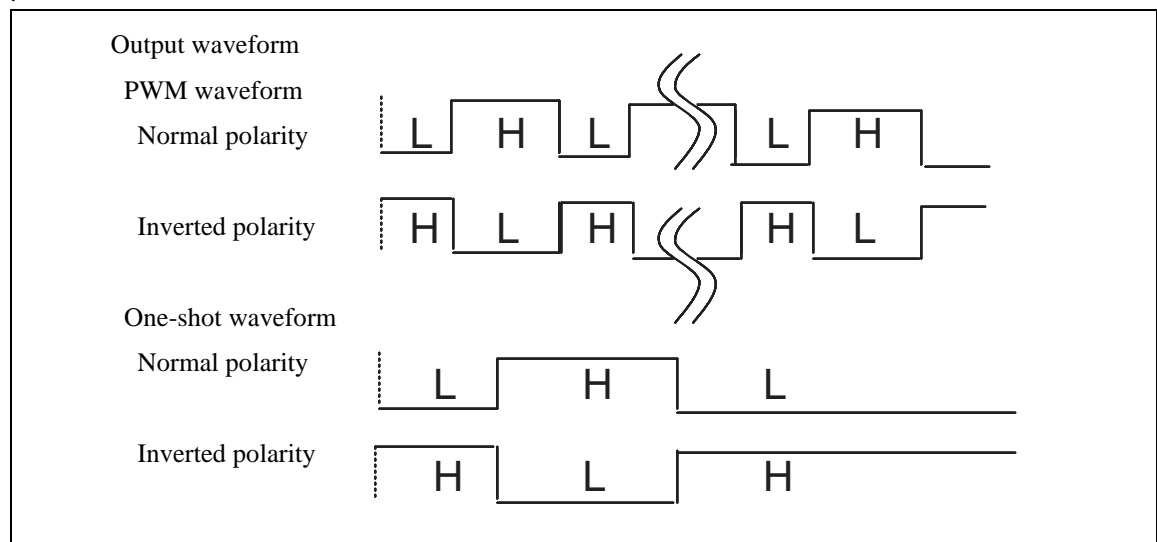
- 17.1 Overview of 16-bit PPG Timer
- 17.2 Configuration of 16-bit PPG Timer
- 17.3 Channels of 16-bit PPG Timer
- 17.4 Pins of 16-bit PPG Timer
- 17.5 Registers of 16-bit PPG Timer
- 17.6 Interrupts of 16-bit PPG Timer
- 17.7 Explanation of 16-bit PPG Timer Operations and Setup Procedure Example
- 17.8 Notes on Using 16-bit PPG Timer
- 17.9 Sample Programs for 16-bit PPG Timer

## 17.1 Overview of 16-bit PPG Timer

The 16-bit PPG timer can generate a PWM (Pulse Width Modulation) output or one-shot (square wave) output, and the period and duty of the output waveform can be changed by software freely. The timer can also generate an interrupt when a start trigger occurs or on the rising or falling edge of the output waveform.

### ■ 16-bit PPG Timer

16-bit PPG timer can output the PWM output and the one shot. The output wave form can be reversed by setting the register (Normal polarity ↔ Inverted polarity)



- The count operation clock can be selected from eight different clock sources (MCLK/1, MCLK/2, MCLK/4, MCLK/8, MCLK/16, MCLK/32,  $F_{CH}/2^7$ , or  $F_{CH}/2^8$ ). (MCLK: Machine clock,  $F_{CH}$ : Main clock)
- Interrupt can be selectively triggered by the following four conditions:
  - Occurrence of a start trigger in the PPG timer
  - Occurrence of a counter borrow in the 16-bit down-counter (cycle match).
  - Rising edge of PPG in normal polarity or falling edge of PPG in inverted polarity
  - Counter borrow, rising edge of PPG in normal polarity, or falling edge of PPG in inverted polarity

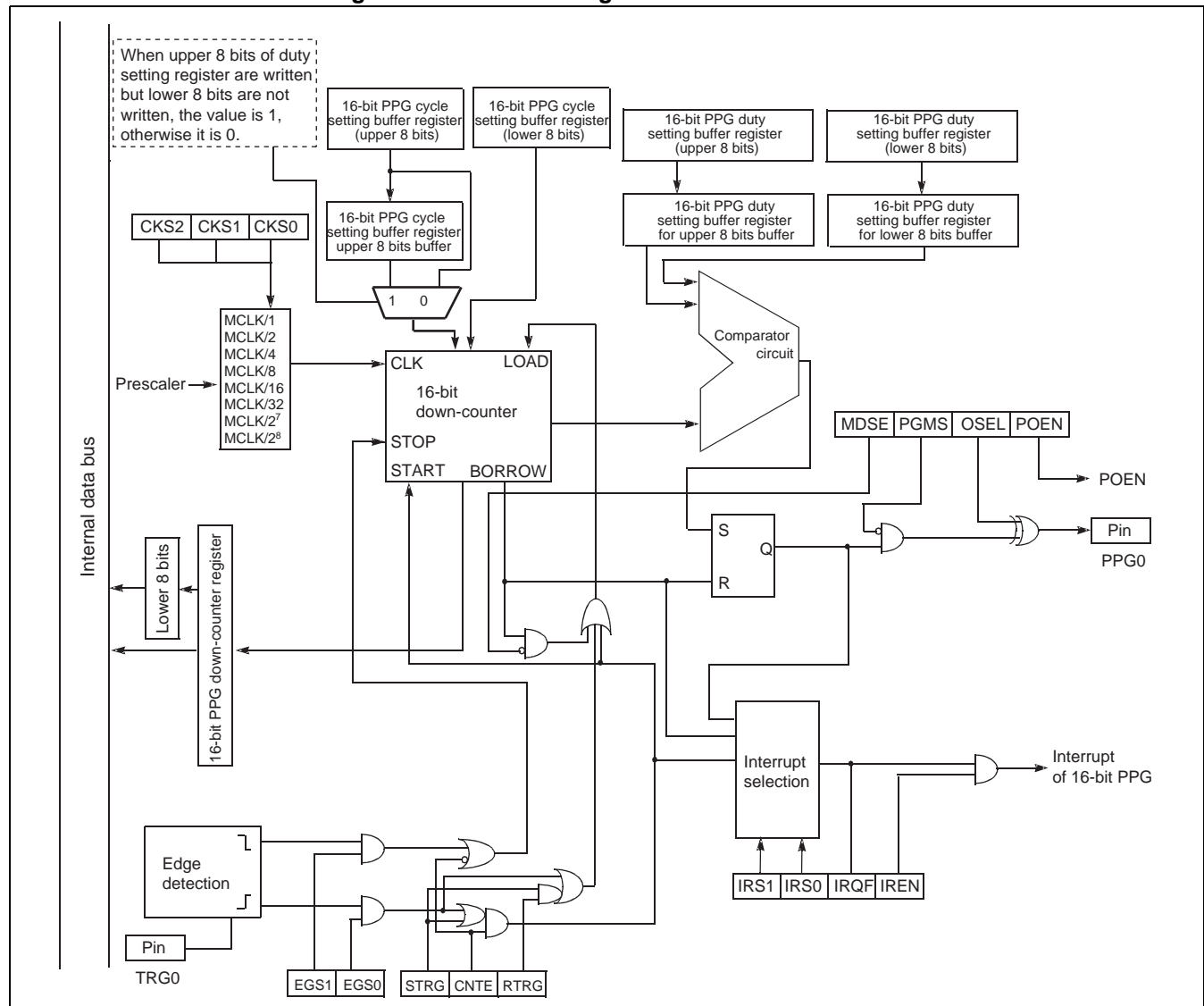
## MB95110B/M Series

### 17.2 Configuration of 16-bit PPG Timer

Shown below is the block diagram of the 16-bit PPG timer.

#### ■ Block Diagram of 16-bit PPG Timer

Figure 17.2-1 Block Diagram of 16-bit PPG Timer



#### ● Count clock selector

The clock for the countdown of 16-bit down-counter is selected from eight types of internal count clocks.

#### ● 16 bit down-counter

It counts down with the count clock selected with the count clock selector.

● **Comparator circuit**

The output is kept "H" until the value of 16-bit down-counter is corresponding to the value of 8/16-bit PPG duty setting buffer register from the value of 16-bit PPG cycle setting buffer register.

Afterwards, after keep "L" the output until the counter value is corresponding to "1", it keeps counting 8-bit down counter from the value of 16-bit PPG cycle setting buffer register.

● **16-bit PPG down-counter register (upper, lower) (PDCRH0, PDCRL0)**

The value of 16-bit down-counter of 16-bit PPG timer is read.

● **16-bit PPG cycle setting buffer register (upper, lower) (PCSRH0, PCSRL0)**

The compare value for the cycle of 16-bit PPG timer is set.

● **16-bit PPG duty setting buffer register (upper, lower) (PDUTH0, PDUTL0)**

The compare value for "H" width of 16-bit PPG timer is set.

● **16-bit PPG status control register (upper, lower) (PCNTH0, PCNTL0)**

The operation mode and the operation condition of 16-bit PPG timer are set.

■ **Input Clock**

The 16-bit PPG timer uses the output clock from the prescaler as its input clock (count clock).

## 17.3 Channels of 16-bit PPG Timer

---

This section describes the channels of the 16-bit PPG timer.

---

### ■ Channels of 16-bit PPG Timer

MB95110B/M series has one 16-bit PPG timer.

Table 17.3-1 and Table 17.3-2 show the correspondence among the channel, pin and register.

**Table 17.3-1 Pins of 16-bit PPG Timer**

Channel	Pin name	Pin function
0	PPG0	PPG0 output
	TRG0	Trigger 0 input

**Table 17.3-2 Registers of 16-bit PPG Timer**

Channel	Register name	Corresponding register (name in this manual)
0	PDCRH0	16-bit PPG down counter register (upper)
	PDCRL0	16-bit PPG down counter register (lower)
	PCSRH0	16-bit PPG cycle setting buffer register (upper)
	PCSRL0	16-bit PPG cycle setting buffer register (lower)
	PDUTH0	16-bit PPG duty setting buffer register (upper)
	PDUTL0	16-bit PPG duty setting buffer register (lower)
	PCNTH0	16-bit PPG status control register (upper)
	PCNTL0	16-bit PPG status control register (lower)

## 17.4 Pins of 16-bit PPG Timer

This section describes the pins of the 16-bit PPG timer.

### ■ Pins of 16-bit PPG Timer

The pin related to the 16-bit PPG timer is namely the PPG0 pin and the TRG0 pin.

#### ● PPG0 pin

Each pin serves as a general-purpose I/O port as well as a 16-bit PPG timer output.

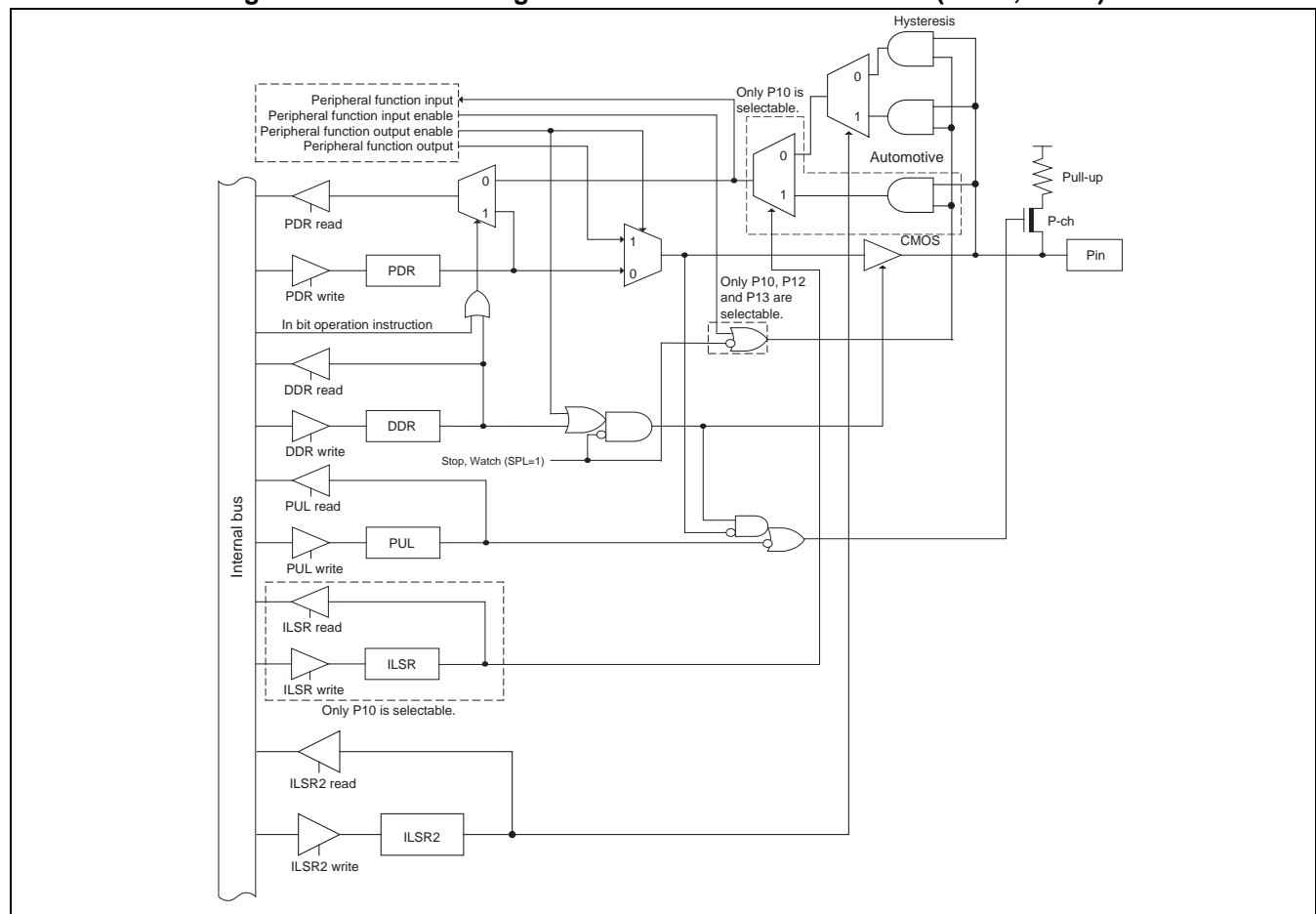
PPG0: A PPG waveform is outputted to these pins. The PPG waveform can be outputted by using the 16-bit PPG status control register to enable output (PCNTL0: POEN=1).

#### ● TRG0 pin

TRG0: Used to start 16-bit PPG timer by hardware trigger.

### ■ Block Diagrams of Pins Related to 16-bit PPG

Figure 17.4-1 Block Diagram of Pin Related to 16-bit PPG (PPG0, TRG0)



## 17.5 Registers of 16-bit PPG Timer

This section describes the registers of the 16-bit PPG timer.

### ■ Registers of 16-bit PPG Timer

Figure 17.5-1 Registers of 16-bit PPG Timer

16-bit PPG down counter register (upper): PDCRH									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAA <sub>H</sub> PDCRH0	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	
16-bit PPG down counter register (lower): PDCRL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAB <sub>H</sub> PDCRL0	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	
16-bit PPG cycle setting buffer register (upper): PCSRH									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAC <sub>H</sub> PCSRH0	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG cycle setting buffer register (lower): PCSRL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAD <sub>H</sub> PCSRL0	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG duty setting buffer register (upper): PDUTH									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAE <sub>H</sub> PDUTH0	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG duty setting buffer register (lower): PDUTL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAF <sub>H</sub> PDUTL0	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG status control register (upper): PCNTH									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0042 <sub>H</sub> PCNTH0	CNTE	STRG	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	00000000 <sub>B</sub>
	R/W	R0,W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG status control register (lower): PCNTL									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0043 <sub>H</sub> PCNTL0	EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL	00000000 <sub>B</sub>
	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									
R/WX: Read only (Readable, writing has no effect on operation)									
R(RM1), W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)									
R0,W: Write-only (Write-only. The read value is "0".)									



## 17.5.1 16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0)

The 16-bit PPG down counter registers (Upper, Lower) (PDCRH0, PDCRL0) form a 16-bit register which is used to read the count value from the 16-bit PPG down-counter.

### ■ 16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0)

Figure 17.5-2 16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0)

16-bit PPG down counter register (upper) PDCRH0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAA <sub>H</sub> PDCRH0	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

16-bit PPG down counter register (lower) PDCRL0

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAB <sub>H</sub> PDCRL0	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX: Read only (Readable, writing has no effect on operation)

These registers form a 16-bit register which is used to read the count value from the 16-bit down-counter. The initial values of the register are all "0".

Always use one of the following procedures to read from this register.

- Use the "MOVW" instruction (use a 16-bit access instruction to read the PDCRH0 register address)
- Use the "MOV" instruction and read PDCRH0 first and PDCRL0 second (reading PDCRH0 automatically copies the lower 8 bits of the down-counter to PDCRL0)

These registers are read-only and writing has no effect on the operation.

Note:

If you use the "MOV" instruction and read PDCRL0 before PDCRH0, PDCRL0 will return the value from the previous valid read operation. Therefore, the value of the 16-bit down-counter will not be read correctly.

## MB95110B/M Series

### 17.5.2 16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0)

The 16-bit PPG cycle setting buffer registers are used to set the cycle for the output pulses generated by the PPG.

#### ■ 16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0)

Figure 17.5-3 16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0)

16-bit PPG cycle setting buffer register (upper) PCSRH0									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAC <sub>H</sub> PCSRH0	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG cycle setting buffer register (lower) PCSRL0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAD <sub>H</sub> PCSRL0	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									

These registers form a 16-bit register which sets the period for the output pulses generated by the PPG. The values set in these registers are loaded to the down-counter.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PCSRH0 register address)
- Use the "MOV" instruction and write to PCSRH0 first and PCSRL0 second  
If a down-counter load occurs after writing data to PCSRH0 (but before writing data to PCSRL0), the previous valid PCSRH0/PCSRL0 value will be loaded to the down-counter. If the PCSRH0/PCSRL0 value is modified during counting, the modified value will become effective from the next load of the down-counter.
- Do not set PCSRH0 and PCSRL0 to "00<sub>H</sub>", or PCSRH0 to "01<sub>H</sub>" and PCSRL0 to "01<sub>H</sub>".

#### Note:

If the down-counter load occurs after the "MOV" instruction is used to write data to PCSRL0 before PCSRH0, the previous valid PCSRH0 value and newly written PCSRL0 value are loaded to the down-counter. It should be noted that as a result, the correct period cannot be set.

### 17.5.3 16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0)

The 16-bit PPG duty setting buffer registers control the duty ratio for the output pulses generated by the PPG.

#### ■ 16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0)

Figure 17.5-4 16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0)

16-bit PPG duty setting buffer register (upper) PDUTH0									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0FAE <sub>H</sub> PDUTH0	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
16-bit PPG duty setting buffer register (lower) PDUTL0									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0FAF <sub>H</sub> PDUTL0	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	11111111 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									

These registers form a 16-bit register which controls the duty ratio for the output pulses generated by the PPG. Transfer of the data from the 16-bit PPG duty setting buffer registers to the duty setting registers is performed at the same timing as the down-counter read.

When writing to these registers, always use one of the following procedures.

- Use the "MOVW" instruction (use a 16-bit access instruction to write to the PDUTH0 register address)
- Use the "MOV" instruction and write to PDUTH0 first and PDUTL0 second  
If a down-counter load occurs after writing data to PDUTH0 (but before writing data to PDUTL0), the value of the 16-bit PPG duty setting buffer registers is not transferred to the duty setting registers.

The relation between the value of the 16-bit PPG duty setting registers and output pulse is as follows:

- When the same value is set in both the 16-bit PPG cycle setting buffer registers and duty setting registers, the "H" level will always be outputted if normal polarity is set, or the "L" level will always be outputted if inverted polarity is set.
- When the duty setting registers are set to "0000<sub>H</sub>", the "L" level will always be outputted if normal polarity is set, or the "H" level will always be outputted if inverted polarity is set.
- When the value set in the duty setting registers is greater than the value in the 16-bit PPG cycle setting buffer registers, the "L" level will always be outputted if normal polarity is set, and the "H" level will always be outputted if inverted polarity is set.

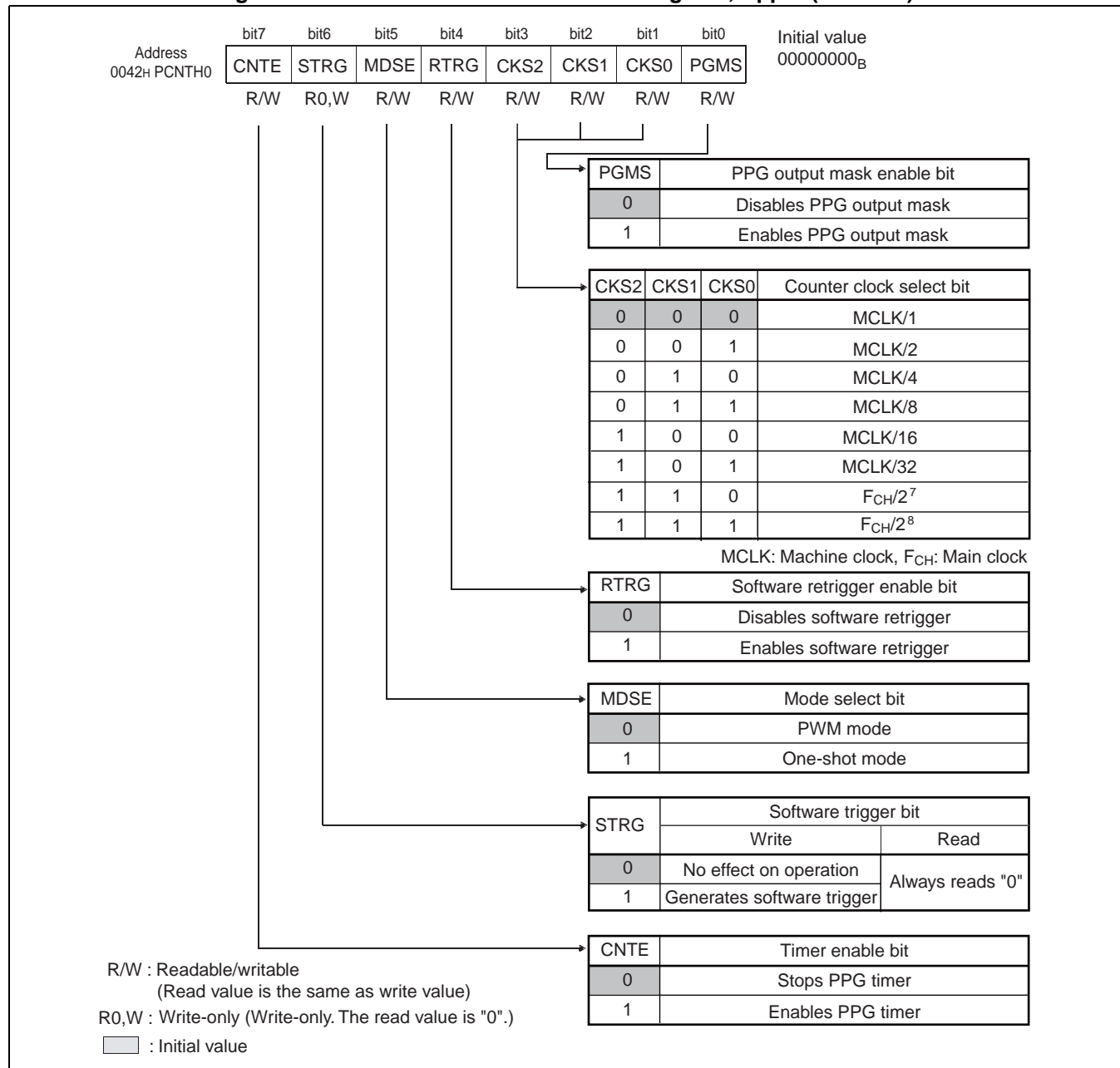
## MB95110B/M Series

### 17.5.4 16-bit PPG Status Control Register (Upper, Lower) (PCNTH0, PCNTL0)

The 16-bit PPG status control register is used to enable and disable the 16-bit PPG timer and also to set the operating status for the software trigger, retrigger control interrupt, and output polarity. This register can also check the operation status.

#### ■ 16-bit PPG Status Control Register, Upper (PCNTH0)

Figure 17.5-5 16-bit PPG Status Control Register, Upper (PCNTH0)

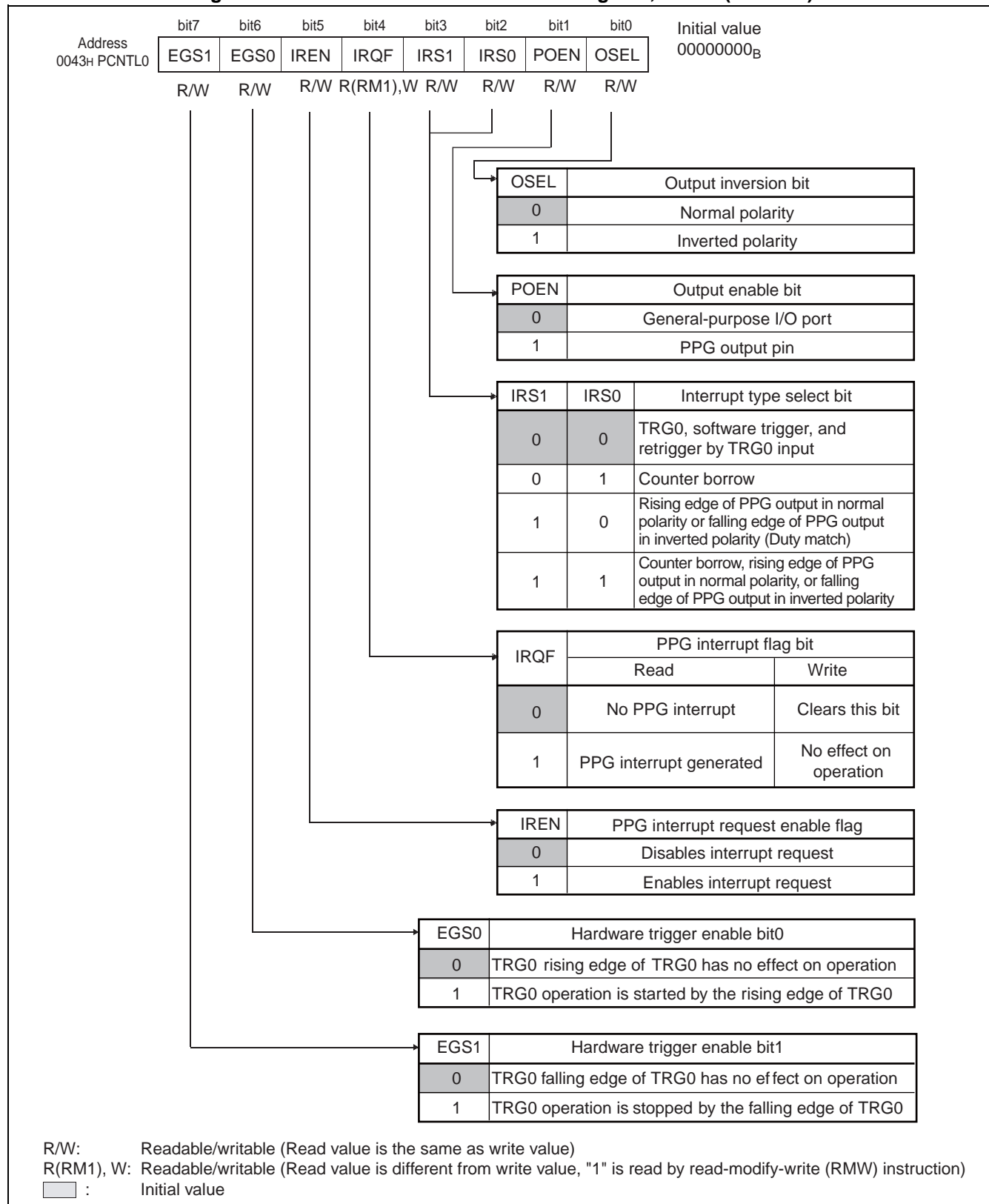


**Table 17.5-1 16-bit PPG Status Control Register, Upper (PCNTH0)**

Bit name		Function
bit7	CNTE: Timer enable bit	This bit is used to enable/stop PPG timer operation. <b>When the bit is set to "0"</b> , the PPG operation halts immediately and the PPG output goes to the initial level ("L" output if OSEL is 0; "H" output if OSEL is 1). <b>When the bit is set to "1"</b> , PPG operation is enabled and the PPG goes to standby to wait for a trigger.
bit6	STRG: Software trigger bit	This bit is used to start the PPG timer by software. <b>When the bit is set to "1"</b> , setting the CNTE bit to "1" starts the PPG timer. Reading this bit always returns "0".
bit5	MDSE: Mode select bit	This bit is used to set the PPG operation mode. <b>When the bit is set to "0"</b> , the PPG operates in PWM mode. <b>When the bit is set to "1"</b> , the PPG operates in one-shot mode. Note: Modifying this bit is prohibited during operation.
bit4	RTRG: Software retrigger enable bit	This bit is used to enable or disable the software retrigger function of the PPG during operation. <b>When the bit is set to "0"</b> , the software retrigger function is "disabled". <b>When the bit is set to "1"</b> , the software retrigger function is "enabled".
bit1, bit2, bit3	CKS2~CKS0: Count clock select bits	These bits select the operating clock for the 16-bit PPG timer. The count clock signal is generated by the prescaler. Refer to "6.12 Operating Explanation of Prescaler". Note: As the time-base timer (TBT) is halted in sub clock mode, $F_{CH}/2^7$ and $F_{CH}/2^8$ cannot be selected in this case.
bit0	PGMS: PPG output mask enable bit	This bit is used to mask the PPG output to a specific level regardless of the mode setting (MDSE: bit5), period setting (PCSRH0, PCSRL0), and duty setting (PDUTH0, PDUTL0). <b>When the bit is set to "0"</b> , the PPG output mask function is disabled. <b>When the bit is set to "1"</b> , the PPG output mask function is enabled. When the PPG output polarity setting is set to "normal" (OSEL bit in PCNTL0 register = 0), the output is always masked to "L". When the polarity setting is set to "inverted" (OSEL bit in PCNTL0 register = 1), the PPG output is always masked to "H".

■ 16-bit PPG Status Control Register, Lower (PCNTL0)

Figure 17.5-6 16-bit PPG Status Control Register, Lower (PCNTL0)



**Table 17.5-2 16-bit PPG Status Control Register, Lower (PCNTL0)**

Bit name		Function															
bit7	EGS1: Hardware trigger enable bit1	This bit determines whether to allow or disallow the falling edge of TRG0 input to stop operation. <b>When the bit is set to "0"</b> , the falling edge of TRG0 has no effect on operation. <b>When the bit is set to "1"</b> , the operation is stopped by the falling edge of TRG0.															
bit6	EGS0: Hardware trigger enable bit0	This bit determines whether to allow or disallow the rising edge of TRG0 input to start operation. <b>When the bit is set to "0"</b> , the rising edge of TRG0 has no effect on operation. <b>When the bit is set to "1"</b> , the operation is started by the rising edge of TRG0.															
bit5	IREN: PPG interrupt request enable bit	This bit enables or disables PPG interrupt request to the interrupt controller. <b>When the bit is set to "0"</b> , an interrupt request is disabled. <b>When the bit is set to "1"</b> , an interrupt request is enabled.															
bit4	IRQF: PPG interrupt flag bit	This bit is set to "1" when a PPG interrupt occurs. <b>When the bit is set to "0"</b> , clears the bit. <b>When the bit is set to "1"</b> , has no effect on operation. "1" is always read in read-modify-write (RMW) instruction.															
bit3, bit2	IRS1, IRS0: Interrupt type select bits	These bits select the interrupt type for the PPG timer. <table border="1"> <thead> <tr> <th>IRS1</th><th>IRS0</th><th>Type of interrupt</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Trigger by TRG0 input, software trigger, or retrigger</td></tr> <tr> <td>0</td><td>1</td><td>Counter borrow</td></tr> <tr> <td>1</td><td>0</td><td>Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr> <tr> <td>1</td><td>1</td><td>Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity</td></tr> </tbody> </table>	IRS1	IRS0	Type of interrupt	0	0	Trigger by TRG0 input, software trigger, or retrigger	0	1	Counter borrow	1	0	Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	1	1	Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity
IRS1	IRS0	Type of interrupt															
0	0	Trigger by TRG0 input, software trigger, or retrigger															
0	1	Counter borrow															
1	0	Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity															
1	1	Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity															
bit1	POEN: Output enable bit	This bit enables or disables output from the PPG output pin. <b>When the bit is set to "0"</b> , the pin serves as a general-purpose port. <b>When the bit is set to "1"</b> , the pin serves as the PPG timer output pin.															
bit0	OSEL: Output inversion bit	This bit selects the polarity of PPG output pin. <b>When the bit is set to "0"</b> , the PPG output goes to "H" when "L" is output in the internal start and the 16-bit down-counter value matches the duty setting register value, and goes to "L" when a down-counter borrow occurs (Normal polarity). <b>When the bit is set to "1"</b> , the PPG output is inverted (Inverted polarity).															

## MB95110B/M Series

### 17.6 Interrupts of 16-bit PPG Timer

The 16-bit PPG timer can generate interrupt requests in the following cases:

- When a trigger or counter borrow occurs
- When a rising edge of PPG is generated in normal polarity
- When a falling edge of PPG is generated in inverted polarity

The interrupt operation is controlled by IRS1 (bit3) and IRS0 (bit2) in the PCNTL0 register.

#### ■ Interrupts of 16-bit PPG Timer

Table 17.6-1 shows interrupt control bits and interrupt sources of the 16-bit PPG timer.

**Table 17.6-1 Interrupt Control Bits and Interrupt Sources of 16-bit PPG Timer**

Item	Description
Interrupt flag bit	PCNTL0:IRQF
Interrupt request enable bit	PCNTL0:IREN
Interrupt type select bits	PCNTL0:IRS1, 0
Interrupt sources	PCNTL0:IRS1, 0=00 Hardware trigger by TRG0 Pin input of 16-bit down-counter, software trigger and retrigger
	PCNTL0:IRS1, 0=01 Counter borrow of 16-bit down-counter
	PCNTL0:IRS1, 0=10 Rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity
	PCNTL0:IRS1, 0=11 Counter borrow of 16-bit down-counter, rising edge of PPG0 output in normal polarity, or falling edge of PPG0 output in inverted polarity

When IRQF (bit4) in the 16-bit PPG status control register (PCNTL0) is set to "1" and interrupt requests are enabled (PCNTL0:IREN: bit5 = 1) in the 16-bit PPG timer, an interrupt request is generated and outputted to the controller.

#### ■ Registers and Vector Table Related to Interrupts of 16-bit PPG Timer

**Table 17.6-2 Registers and Vector Table Related to Interrupts of 16-bit PPG Timer**

Interrupt source	Interrupt request No.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.0	IRQ15	ILR3	L15	FFDC <sub>H</sub>	FFDD <sub>H</sub>

ch: Channel

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.



## 17.7 Explanation of 16-bit PPG Timer Operations and Setup Procedure Example

---

**The 16-bit PPG timer can operate in PWM mode or one-shot mode. In addition, a retrigger function can be used in the 16-bit PPG timer.**

---

### ■ PWM Mode (MDSE of PCNTH Register: bit5 = 0)

In PWM operation mode, the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are loaded and the 16-bit down-counter starts down-count operation when a software trigger is inputted or a hardware trigger by TRG0 pin input is inputted. When the count value reaches "1", the 16-bit PPG cycle setting buffer register (PCSRH0, PCSRL0) values are reloaded to repeat the down-count operation.

The initial state of the PPG output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H" synchronizing with count clock. The output changes back to "L" when the "H" was output until the value of duty setting. (The output levels will be reversed if OSEL is set to "1".)

When the retrigger function is disabled (RTRG = 0), software triggers (STRG = 1) are ignored during the operation of the down-counter.

When the down-counter is not running, the maximum time between a valid trigger input occurring and the down-counter starting is as follows.

Software trigger: 1 count clock cycle + 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 1 count clock cycle + 3 machine clock cycles

The minimum time is as follows.

Software trigger: 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 3 machine clock cycles

When the down-counter is running, the maximum time between a valid retrigger input occurring and the down-counter restarting is as follows.

Software trigger: 1 count clock cycle + 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 1 count clock cycle + 3 machine clock cycles

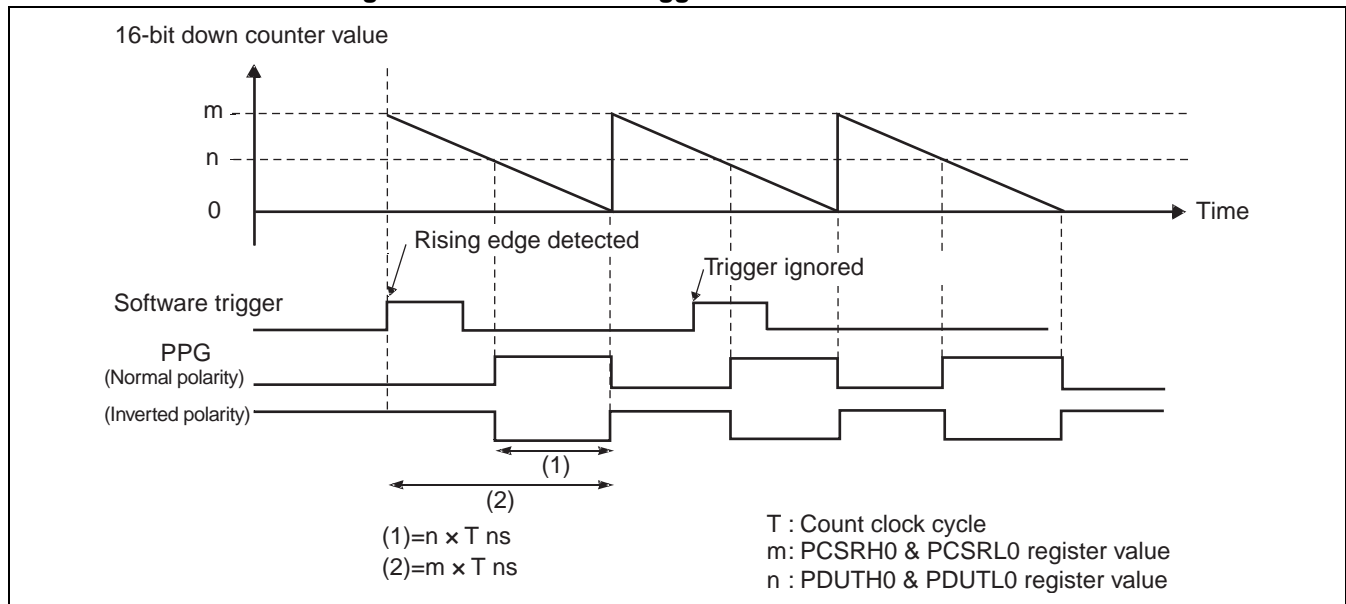
The minimum time is as follows.

Software trigger: 2 machine clock cycles

Hardware trigger by TRG0 Pin input: 3 machine clock cycles

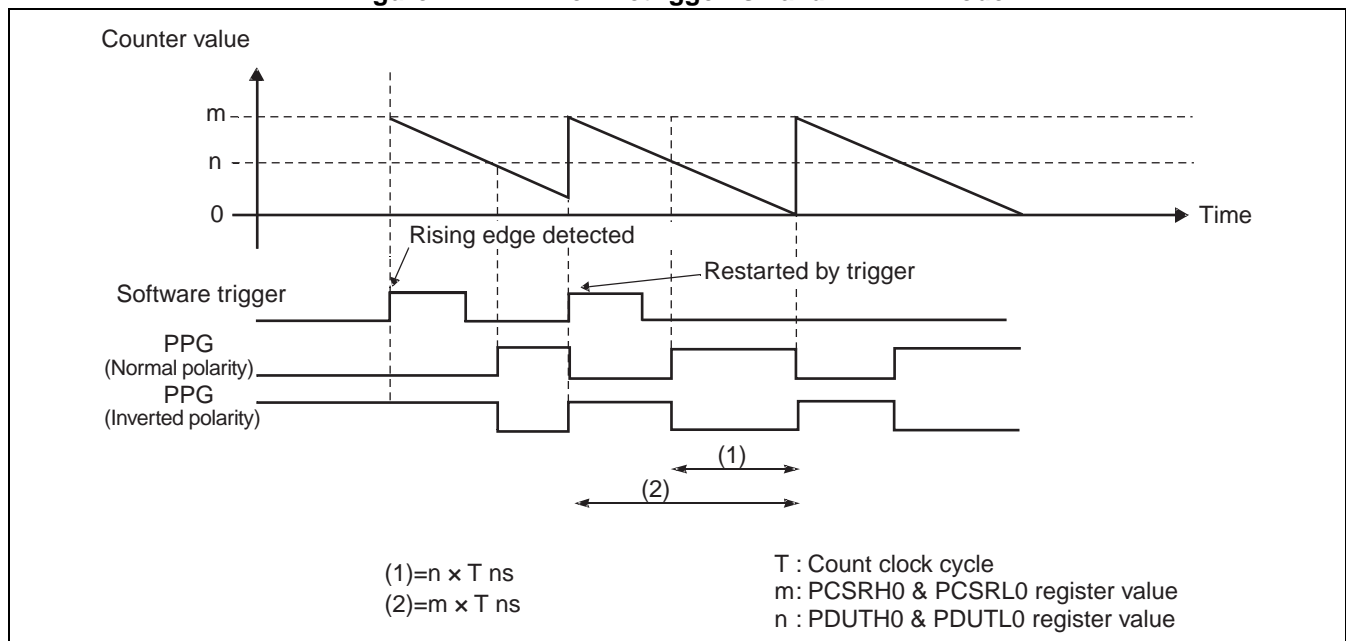
- Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 17.7-1 When Retrigger Is Invalid in PWM Mode



- Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 17.7-2 When Retrigger Is Valid in PWM Mode



## Example

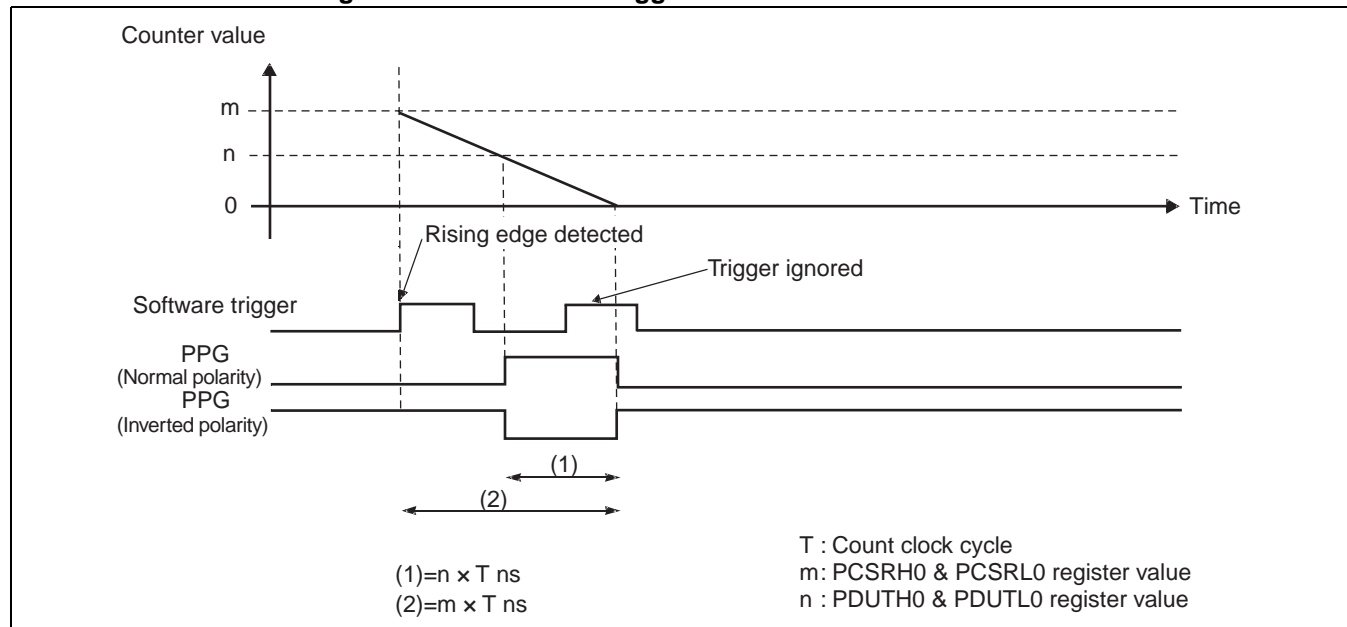
## ■ One-shot Mode (MDSE of PCNTH0 Register: bit5 = 1)

One-shot operation mode can be used to output a single pulse with a specified width when a valid trigger input occurs. When retriggering is enabled and a valid trigger is detected during the counter operation, the down counter value is reloaded.

The initial state of the PPG output is "L". When the 16-bit down-counter value matches the value set in the duty setting registers, the output changes to "H". The output changes back to "L" when the counter reaches "1". (The output levels will be reversed if OSEL is set to 1.)

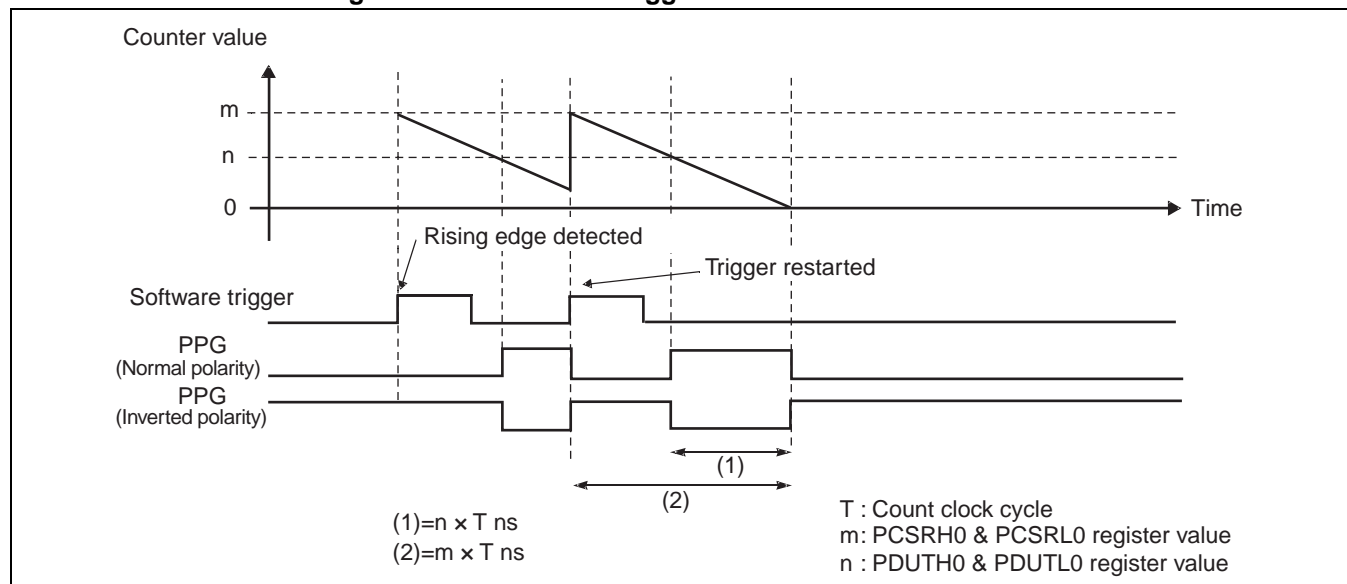
## ● Invalidating the retrigger (RTRG of PCNTH0 register: bit4 = 0)

Figure 17.7-3 When Retrigger Is Invalid in One-shot Mode



## ● Validating the retrigger (RTRG of PCNTH0 register: bit4 = 1)

Figure 17.7-4 When Retrigger Is Valid in One-shot Mode



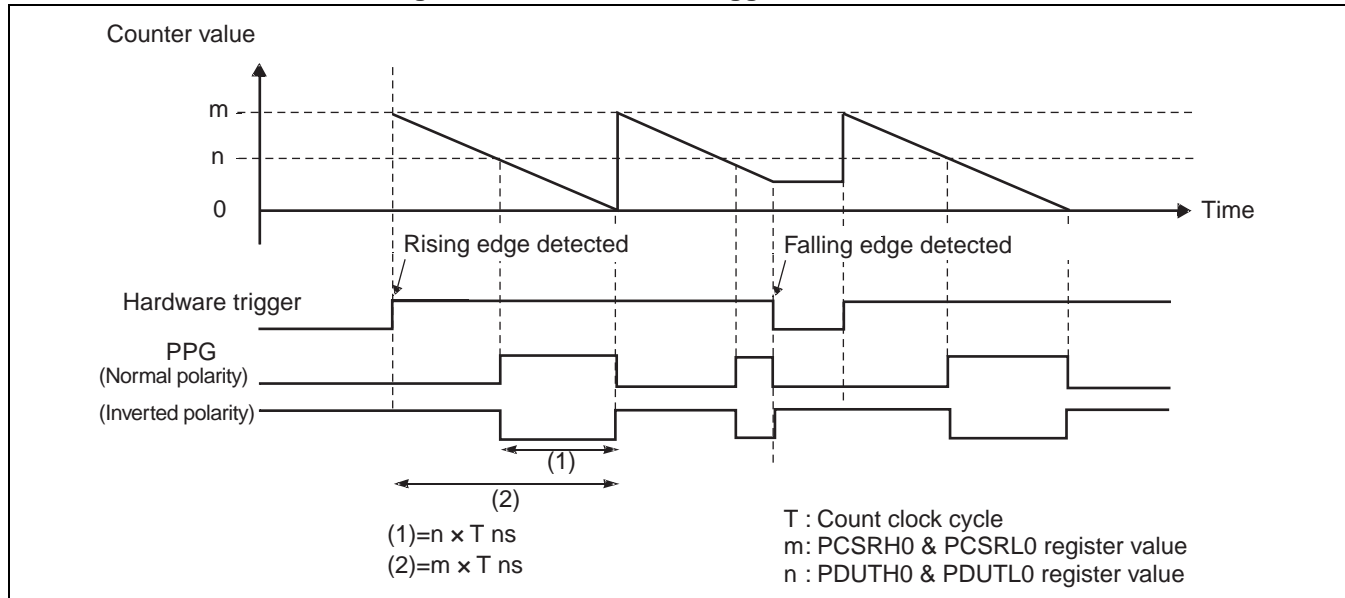
## ■ Hardware Trigger

"Hardware trigger" refers to PPG activation by signal input to the TRG0 input pin. When EGS1 and EGS0 are set to "11<sub>B</sub>" and the hardware trigger is used with TRG0 input, PPG starts operation on a rising edge and halts the operation upon the detection of a falling edge.

Moreover, the PPG timer begins operation of the following rising edge from the beginning.

The operation can be retriggered by a valid TRG0 input hardware trigger regardless of the retrigger setting of the RTRG bit when the TRG0 input hardware trigger has been selected.

**Figure 17.7-5 Hardware Trigger in PWM Mode**



## ■ Setup Procedure Example

The 16-bit PPG timer is set up in the following procedure:

### ● Initial setting

- 1) Set the interrupt level (ILR3, ILR4)
- 2) Enable the hardware trigger and interrupts, select the interrupt type, and enable output (PCNTL)
- 3) Select the count clock and the mode, and enable timer operation (PCNTH)
- 4) Set the cycle (PCSRL0, PCSRH0)
- 5) Set the duty (PDUTH0, PDUTL0)
- 6) Start the PPG by the software trigger (PCNTH:STRG = 1)

### ● Interrupt processing

- 1) Process any interrupt
- 2) Clear the interrupt request flag (PCNTL:IRQF)

## 17.8 Notes on Using 16-bit PPG Timer

---

**Shown below are the precautions that must be followed when using the 16-bit PPG timer.**

---

### ■ Notes on Using 16-bit PPG Timer

#### ● Precautions when setting the program

Do not use the retrigger if the same values are set for the cycle and duty. If used, the PPG output will go to the "L" level for one count clock cycle after the retrigger, and then go back to the "H" level when normal polarity has been selected.

If the microcontroller enters a standby mode, the TRG0 pin setting may change and cause the device to malfunction. Therefore, disable the timer enable bit (PCNTH0:CNTE = 0) or disable the hardware trigger enable bit (PCNTL0:EGS1, EGS0 = 00<sub>B</sub>).

When the cycle and duty are set to the same value, an interrupt is generated only once by duty match. Moreover, if the duty is set to a value greater than the value of the period, no interrupt will be generated by duty match.

Do not disable the timer enable bit (PCNTH0: CNTE = 0) and software trigger (PCNTH0: STRG =1) at the same time when retrigger by the software is enabled (PCNTH0: RTRG =1) and the retrigger is selected as an interrupt type (PCNTL0: IRS1, IRS0 = 00<sub>B</sub>) during count operation. If it occurs, interrupt flag bit may set by retrigger although timer stops.

## 17.9 Sample Programs for 16-bit PPG Timer

We provide sample programs that can be used to operate the 16-bit PPG timer.

### ■ Sample Programs for 16-bit PPG Timer

For information about the sample programs for the 16-bit PPG timer, refer to "■ Sample Programs" in Preface.

### ■ Setup Methods without Sample Program

#### ● How to set the PPG operation mode

The operation mode select bit (PCNTH0.MDSE) is used.

Operation mode	Operation mode select bit (MDSE)
PWM mode	Set the bit to "0"
One-shot mode	Set the bit to "1"

#### ● How to select the operating clock

The operating clock select bits (PCNTH0.CKS2/CKS1/CKS0) are used to select the clock.

#### ● How to enable/disable the PPG output pin

The output enable bit (PCNTL0.POEN) is used.

What to be controlled	Output enable bit (POEN)
When enabling PPG output	Set the bit to "1"
When disabling PPG output	Set the bit to "0"

#### ● How to enable/disable PPG operation

The timer enable bit (PCNTH0.CNTE) is used.

What to be controlled	Timer enable bit (CNTE)
When disabling PPG operation	Set the bit to "0"
When enabling PPG operation	Set the bit to "1"

Enable PPG operation before starting the PPG.

● How to start PPG operation by software

The software trigger bit (PCNTH0.STRG) is used.

What to be controlled	Software trigger bit (STRG)
When starting PPG operation by software	Set the bit to "1"

● How to enable/disable the retrigger function of the software trigger

The retrigger enable bit (PCNTL0.RTRG) is used.

What to be controlled	Retrigger enable bit (RTRG)
When enabling retrigger function	Set the bit to "1"
When disabling retrigger function	Set the bit to "0"

● How to start/stop operation on a rising edge of trigger input

The hardware trigger enable bit (PCNTL0.EGS0) is used.

What to be controlled	Hardware trigger enable bit (EGS0)
When starting operation on rising edge	Set the bit to "1"
When stopping operation on rising edge	Set the bit to "0"

● How to start/stop operation on a falling edge of trigger input

The hardware trigger enable bit (PCNTH0.EGS1) is used.

What to be controlled	Hardware trigger enable bit (EGS1)
When starting operation on falling edge	Set the bit to "1"
When stopping operation on falling edge	Set the bit to "0"

● How to invert PPG output

The output inversion bit (PCNTL0.OSEL) is used.

What to be controlled	Output inversion bit (OSEL)
When inverting PPG output	Set the bit to "1"

## ● How to set the PPG output to the "H" or "L" level

The PPG output mask enable bit (PCNTH0.PGMS) and the output inversion bit (PCNTL0.OSEL) are used.

What to be controlled	PPG output mask enable bit (PGMS)	Output inversion bit (OSEL)
When setting output to "H" level	Set the bit to "1"	Set the bit to "1"
When setting output to "L" level	Set the bit to "1"	Set the bit to "0"

## ● How to select the interrupt source

The interrupt select bits (PCNTL0.IRS1/IRS0) are used to select the interrupt source.

Interrupt source	Interrupt select bits (IRS1/IRS0)
Trigger by TRG0 input, software trigger, or retrigger	Set the bits to "00 <sub>B</sub> "
Counter borrow	Set the bits to "01 <sub>B</sub> "
Rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	Set the bits to "10 <sub>B</sub> "
Counter borrow, rising edge of PPG output in normal polarity, or falling edge of PPG output in inverted polarity	Set the bits to "11 <sub>B</sub> "

## ● Interrupt-related registers

The interrupt level is set by the level setting registers shown in the following table.

Interrupt source	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR3) Address: 0007C <sub>H</sub>	#15 Address: 0FFDC <sub>H</sub>

## ● How to enable/disable/clear interrupts

The interrupt request enable bit (PCNTL0.IREN) is used to enable interrupts.

What to be controlled	Interrupt request enable bit (IREN)
When disabling interrupt request	Set the bit to "0"
When enabling interrupt request	Set the bit to "1"

The interrupt request flag (PCNTL0.IRQF) is used to clear interrupt requests.

What to be controlled	Interrupt request flag (IRQF)
When clearing interrupt request	Write "0" to the bit





# **CHAPTER 18**

---

# ***EXTERNAL INTERRUPT CIRCUIT***

**This chapter describes the functions and operations of the external interrupt circuit.**

- 18.1 Overview of External Interrupt Circuit
- 18.2 Configuration of External Interrupt Circuit
- 18.3 Channels of External Interrupt Circuit
- 18.4 Pins of External Interrupt Circuit
- 18.5 Registers of External Interrupt Circuit
- 18.6 Interrupts of External Interrupt Circuit
- 18.7 Explanation of External Interrupt Circuit Operations and Setup Procedure Example
- 18.8 Notes on Using External Interrupt Circuit
- 18.9 Sample Programs for External Interrupt Circuit

## **18.1 Overview of External Interrupt Circuit**

---

**The external interrupt circuit detects edges on the signal that is inputted to the external interrupt pin and generates interrupt requests to the CPU.**

---

### **■ Functions of External Interrupt Circuit**

The external interrupt circuit has the functions to detect any edge of a signal that is inputted to an external interrupt pin and generate an interrupt request to the CPU. This interrupt allows the unit to recover from a standby mode and return to its normal operation.

## 18.2 Configuration of External Interrupt Circuit

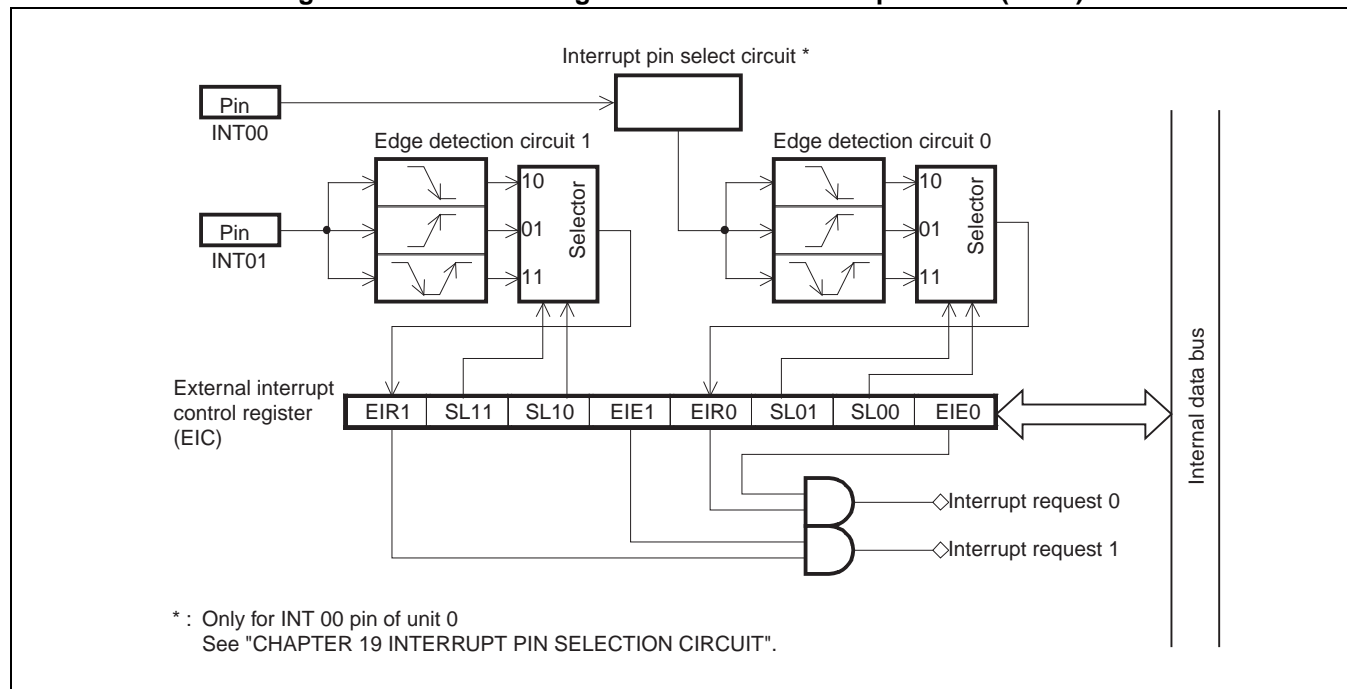
The external interrupt circuit consists of the following blocks:

- Edge detection circuit
- External interrupt control register

### ■ Block Diagram of External Interrupt Circuit

Figure 18.2-1 shows the block diagram of the external interrupt circuit.

**Figure 18.2-1 Block Diagram of External Interrupt Circuit (Unit0)**



#### ● Edge detection circuit

When the polarity of the edge detected on a signal inputted to an external interrupt circuit pin (INT) matches the polarity of the edge selected in the interrupt control register (EIC), the corresponding external interrupt request flag bit (EIR) is set to "1".

#### ● External interrupt control register (EIC)

This register is used to select the valid edge, enable or disable interrupt requests, check for interrupt requests, etc.

## 18.3 Channels of External Interrupt Circuit

This section describes the channels of the external interrupt circuit.

### ■ Channels of External Interrupt Circuit

In MB95110B/M series, each unit has four channels of the external interrupt circuit.

Table 18.3-1 and Table 18.3-2 show the correspondence among the channel, pin and register.

**Table 18.3-1 Pins of External Interrupt Circuit**

Unit	Pin name	Pin function
0	INT00	External interrupt input ch.0
	INT01	External interrupt input ch.1
1	INT02	External interrupt input ch.2
	INT03	External interrupt input ch.3
2	INT04	External interrupt input ch.4
	INT05	External interrupt input ch.5
3	INT06	External interrupt input ch.6
	INT07	External interrupt input ch.7

**Table 18.3-2 Registers of External Interrupt Circuit**

Unit	Register name	Corresponding register (Name in this manual)
0	EIC00	EIC: External Interrupt Control register
1	EIC10	
2	EIC20	
3	EIC30	

The following sections only describe the unit 0 side of the external interrupt circuit.

The other units are the same as the unit 0 side of the external interrupt circuit.

**This section shows the pins related to the external interrupt circuit and the block diagram of such pins.**

The pins related to the external interrupt circuit are the INT00 to INT07 pins.

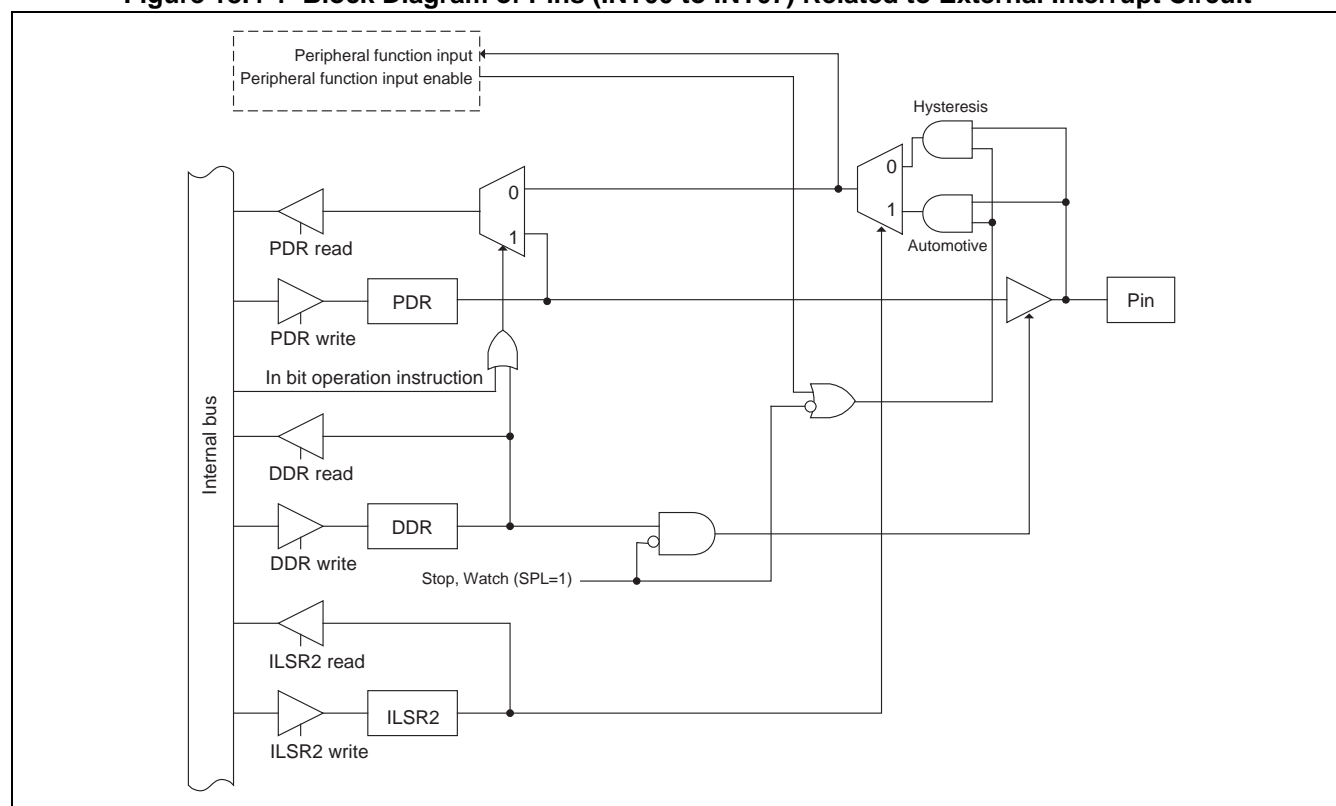
These pins serve both as external interrupt inputs and as general-purpose I/O ports.

INT00 to INT07: When the corresponding pin of the INT00 to INT07 pins is set as an input port by the port direction register (DDR) and the corresponding external interrupt input is enabled by the external interrupt control register (EIC), that pin functions as an external interrupt input pin (INT00 = INT07).

The state of pins can be read from the port data register (PDR) whenever input port is set as a pin function. However, the value of PDR is read when read-modify-write (RMW) instruction is used.

### ■ Block Diagram of Pins Related to External Interrupt Circuit

**Figure 18.4-1 Block Diagram of Pins (INT00 to INT07) Related to External Interrupt Circuit**



## 18.5 Registers of External Interrupt Circuit

This section describes the registers of the external interrupt circuit.

### ■ List of Registers of External Interrupt Circuit

Figure 18.5-1 shows the registers of the external interrupt circuit.

**Figure 18.5-1 Registers of External Interrupt Circuit**

External interrupt control register (EIC)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0048 <sub>H</sub> EIC00	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000 <sub>B</sub>
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0049 <sub>H</sub> EIC10	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000 <sub>B</sub>
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004A <sub>H</sub> EIC20	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000 <sub>B</sub>
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
004B <sub>H</sub> EIC30	EIR1	SL11	SL10	EIE1	EIR0	SL01	SL00	EIE0	00000000 <sub>B</sub>
	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									
R(RM1), W: Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)									

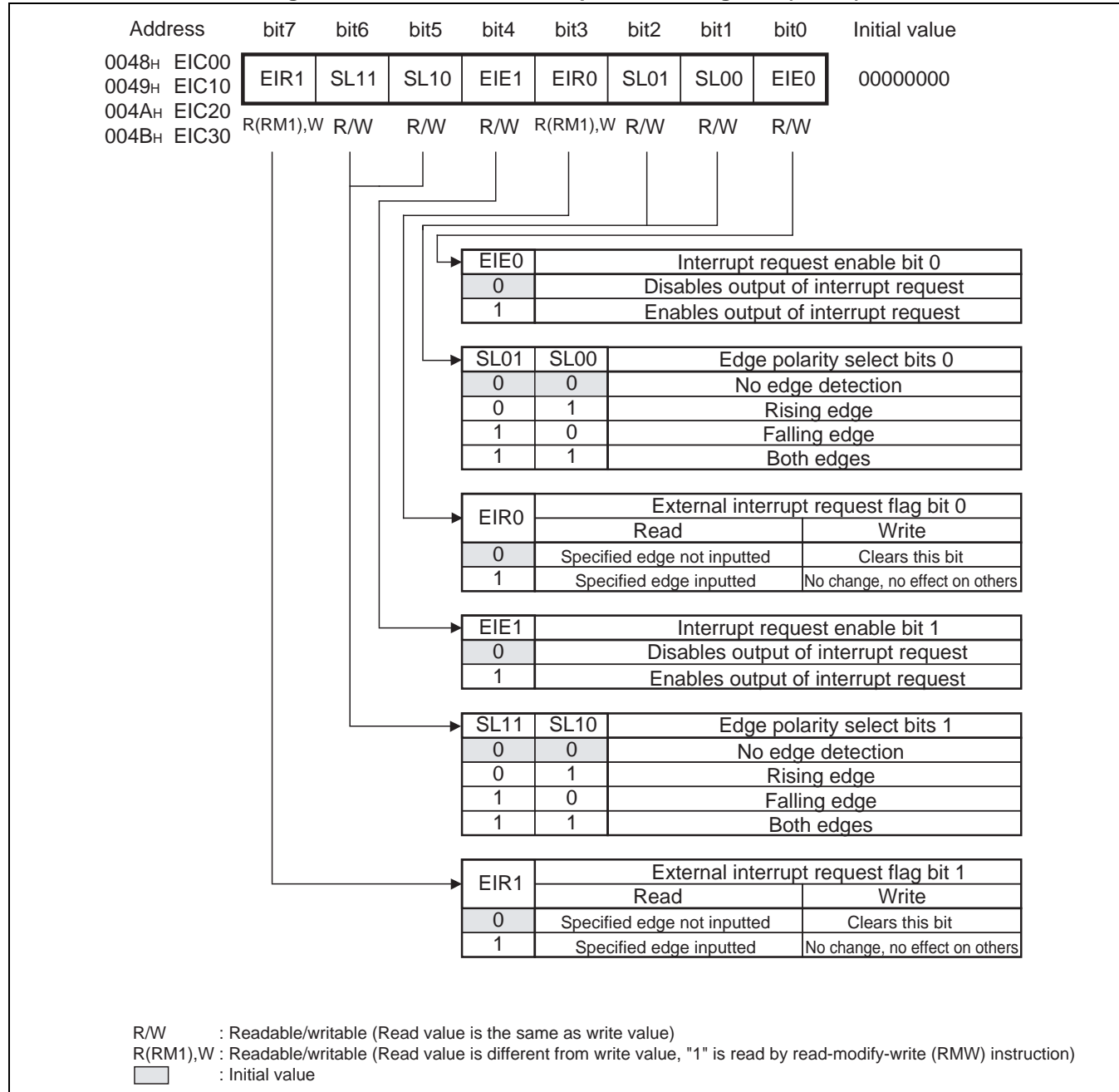
## MB95110B/M Series

### 18.5.1 External Interrupt Control Register (EIC00)

The external interrupt control register (EIC00) is used to select the edge polarity for the external interrupt input and control interrupts.

#### ■ External Interrupt Control Register (EIC00)

Figure 18.5-2 External Interrupt Control Register (EIC00)





**Table 18.5-1 Functional Description of Each Bit of External Interrupt Control Register (EIC00)**

Bit name		Function
bit7	EIR1: External interrupt request flag bit 1	<p>This flag is set to "1" when the edge selected by the edge polarity select bits (SL11, SL10) is inputted to the external interrupt pin INT01.</p> <ul style="list-style-type: none"> <li>When this bit and the interrupt request enable bit 1 (EIE1) are set to "1", an interrupt request is outputted.</li> <li>Writing "0" clears the bit. Writing "1" has no effect.</li> <li>"1" is read in read-modify-write (RMW) instructions.</li> </ul>
bit6, bit5	SL11, SL10: Edge polarity select bits 1	<p>These bits select the polarity of the interrupt source edge of the pulse inputted to the external interrupt pin INT01.</p> <ul style="list-style-type: none"> <li>Edge detection is not performed and no interrupt is generated when these bits are set to "00<sub>B</sub>".</li> <li>Rising edges are detected when these bits are "01<sub>B</sub>", falling edges when "10<sub>B</sub>", and both edges when "11<sub>B</sub>".</li> </ul>
bit4	EIE1: Interrupt request enable bit 1	<p>This bit is used to enable and disable output of interrupt requests to the interrupt controller. When this bit and the external interrupt request flag bit 1 (EIR1) are "1", an interrupt request is outputted.</p> <ul style="list-style-type: none"> <li>When using an external interrupt pin, write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input.</li> <li>The status of the external interrupt pin can be read directly from the port data register, regardless of the status of the interrupt request enable bit.</li> </ul>
bit3	EIR0: External interrupt request flag bit 0	<p>This flag is set to "1" when the edge selected by the edge polarity select bits (SL01, SL00) is inputted to the external interrupt pin INT00.</p> <ul style="list-style-type: none"> <li>When this bit and the interrupt request enable bit 0 (EIE0) are set to "1", an interrupt request is outputted.</li> <li>Writing "0" clears the bit. Writing "1" has no effect.</li> <li>"1" is read in read-modify-write (RMW) instructions.</li> </ul>
bit2, bit1	SL01, SL00: Edge polarity select bits 0	<p>These bits are used to select the polarity of the interrupt source edge of the pulse inputted to the external interrupt pin INT00.</p> <ul style="list-style-type: none"> <li>Edge detection is not performed and no interrupt request is generated when these bits are "00<sub>B</sub>".</li> <li>Rising edges are detected when the bits are "01<sub>B</sub>", falling edges when "10<sub>B</sub>", and both edges when "11<sub>B</sub>".</li> </ul>
bit0	EIE0: Interrupt request enable bit 0	<p>This bit enables or disables the output of interrupt requests to the interrupt controller. An interrupt request is outputted when this bit and the external interrupt request flag bit 0 (EIR0) are "1".</p> <ul style="list-style-type: none"> <li>When using an external interrupt pin, write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input.</li> <li>The status of the external interrupt pin can be read directly from the port data register (PFR), regardless of the status of the interrupt request enable bit.</li> </ul>

## 18.6 Interrupts of External Interrupt Circuit

The interrupt sources for the external interrupt circuit include detection of the specified edge of the signal inputted to an external interrupt pin.

### ■ Interrupt During Operation of External Interrupt Circuit

When the specified edge of external interrupt input is detected, the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1". In this case, an interrupt request will be generated to the interrupt controller, if the corresponding interrupt request enable bit is enabled (EIC: EIE0, EIE1=1). Write "0" to the corresponding external interrupt request flag bit to clear the interrupt request in the interrupt process routine.

### ■ Registers and Vector Table Related to Interrupts of External Interrupt Circuit

**Table 18.6-1 Registers and Vector Table Related to Interrupts of External Interrupt Circuit**

Interrupt source	Interrupt request No.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.0	IRQ0	ILR0	L00	FFFA <sub>H</sub>	FFFB <sub>H</sub>
ch.4					
ch.1	IRQ1	ILR0	L01	FFF8 <sub>H</sub>	FFF9 <sub>H</sub>
ch.5					
ch.2	IRQ2	ILR0	L02	FFF6 <sub>H</sub>	FFF7 <sub>H</sub>
ch.6					
ch.3	IRQ3	ILR0	L03	FFF4 <sub>H</sub>	FFF5 <sub>H</sub>
ch.7					

ch.: Channel

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

## 18.7 Explanation of External Interrupt Circuit Operations and Setup Procedure Example

This section describes the operation of the external interrupt circuit.

### ■ Operation of External Interrupt Circuit

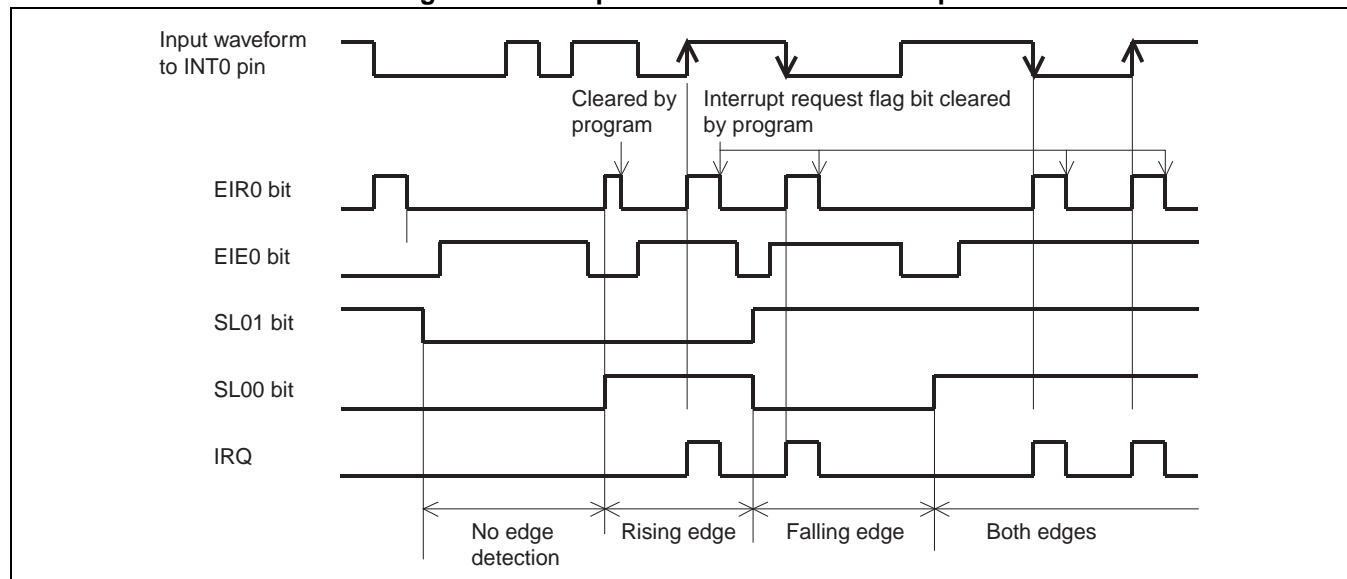
When the polarity of an edge of a signal inputted from one of the external interrupt pins (INT0, 1) matches the polarity of the edge selected by the external interrupt control register (EIC: SL00, SL01, SL10, SL11), the corresponding external interrupt request flag bit (EIC: EIR0, EIR1) is set to "1" and the interrupt request is generated.

Always set the interrupt enable bit to "0" when not using an external interrupt to recover from a standby mode.

When setting the edge polarity select bit (SL), set the interrupt request enable bit (EIE) to "0" to prevent the interrupt request from being generated accidentally. Also clear the interrupt request flag bit (EIR) to "0" after changing the edge polarity.

Figure 18.7-1 shows the operation for setting the INT0 pin as an external interrupt input.

Figure 18.7-1 Operation of External Interrupt



**■ Setup Procedure Example**

The external interrupt circuit is set up in the following procedure:

**● Initial setting**

- 1) Set the interrupt level. (ILR0)
- 2) Select the edge polarity. (EIC:SL01, SL00)
- 3) Enable interrupt requests. (EIC:EIE0 = 1)

**● Interrupt processing**

- 1) Clear the interrupt request flag. (EIC:EIR0 = 0)
- 2) Process any interrupt.

---

**Note:**

The external interrupt input is also used as an I/O port. Therefore, when it is used as the external interrupt input, the corresponding bit in the port direction register (DDR) must be set to "0" (input).

---

## **18.8 Notes on Using External Interrupt Circuit**

---

**This section describes the precautions that must be followed when using the external interrupt circuit.**

---

### **■ Notes on Using External Interrupt Circuit**

- Set the interrupt request enable bit (EIE) to "0" (disabling interrupt requests) when setting the edge polarity select bit (SL). Also clear the external interrupt request flag bit (EIR) to "0" after setting the edge polarity.
- The operation cannot recover from the interrupt processing routine if the external interrupt request flag bit is "1" and the interrupt request enable bit is enabled. Always clear the external interrupt request flag bit in the interrupt processing routine.

## 18.9 Sample Programs for External Interrupt Circuit

We provide sample programs that can be used to operate the external interrupt circuit.

### ■ Sample Programs for External Interrupt Circuit

For information about the sample programs for the external interrupt circuit, refer to "■ Sample Programs" in Preface.

### ■ Setup Methods without Sample Program

#### ● Detection levels and setup methods

Four detection levels are available: no edge detection, rising edge, falling edge, both edges

The detection level bits (EIC: SL01, SL00 or EIC: SL11, SL10) are used.

Operation mode	Detection level bits (SL01, SL00 or SL11, SL10)
No edge detection	Set "00 <sub>B</sub> "
Detecting rising edges	Set "01 <sub>B</sub> "
Detecting falling edges	Set "10 <sub>B</sub> "
Detecting both edges	Set "11 <sub>B</sub> "

#### ● How to use the external interrupt pin

Set the corresponding data direction register (DDR0) to "0".

Operation	Direction bit (P00 to P07)	Setting
Using INT00 pin for external interrupt	DDR0. P00	Set the register to "0"
Using INT01 pin for external interrupt	DDR0. P01	Set the register to "0"
Using INT02 pin for external interrupt	DDR0. P02	Set the register to "0"
Using INT03 pin for external interrupt	DDR0. P03	Set the register to "0"
Using INT04 pin for external interrupt	DDR0. P04	Set the register to "0"
Using INT05 pin for external interrupt	DDR0. P05	Set the register to "0"
Using INT06 pin for external interrupt	DDR0. P06	Set the register to "0"
Using INT07 pin for external interrupt	DDR0. P07	Set the register to "0"

● Interrupt-related registers

The interrupt level is set by the interrupt level setting registers shown in the following table.

Channel	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#0 Address: 0FFFA <sub>H</sub>
ch.1	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#1 Address: 0FFF8 <sub>H</sub>
ch.2	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#2 Address: 0FFF6 <sub>H</sub>
ch.3	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#3 Address: 0FFF4 <sub>H</sub>
ch.4	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#0 Address: 0FFFA <sub>H</sub>
ch.5	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#1 Address: 0FFF8 <sub>H</sub>
ch.6	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#2 Address: 0FFF6 <sub>H</sub>
ch.7	Interrupt level register (ILR0) Address: 00079 <sub>H</sub>	#3 Address: 0FFF4 <sub>H</sub>

● How to enable/disable/clear interrupts

Interrupts are enabled by the interrupt enable bit (EIC00:EIE0 or EIC00:EIE1).

Operation	Interrupt enable bit (EIE0 or EIE1)
When disabling interrupt request	Set the bit to "0"
When enabling interrupt request	Set the bit to "1"

Interrupt requests are cleared by the interrupt request bit (EIC00:EIR0 or EIC00:EIR1).

Operation	Interrupt request bit (EIR0 or EIR1)
When clearing interrupt request	Write "0"

# ***CHAPTER 19***

---

# ***INTERRUPT PIN SELECTION CIRCUIT***

**This chapter describes the functions and operations of the interrupt pin selection circuit.**

- 19.1 Overview of Interrupt Pin Selection Circuit
- 19.2 Configuration of Interrupt Pin Selection Circuit
- 19.3 Pins of Interrupt Pin Selection Circuit
- 19.4 Registers of Interrupt Pin Selection Circuit
- 19.5 Operating Description of Interrupt Pin Selection Circuit
- 19.6 Notes on Using Interrupt Pin Selection Circuit



## 19.1 Overview of Interrupt Pin Selection Circuit

---

**The interrupt pin selection circuit selects pins to be used as interrupt input pins from among various peripheral input pins.**

---

### ■ Interrupt Pin Selection Circuit

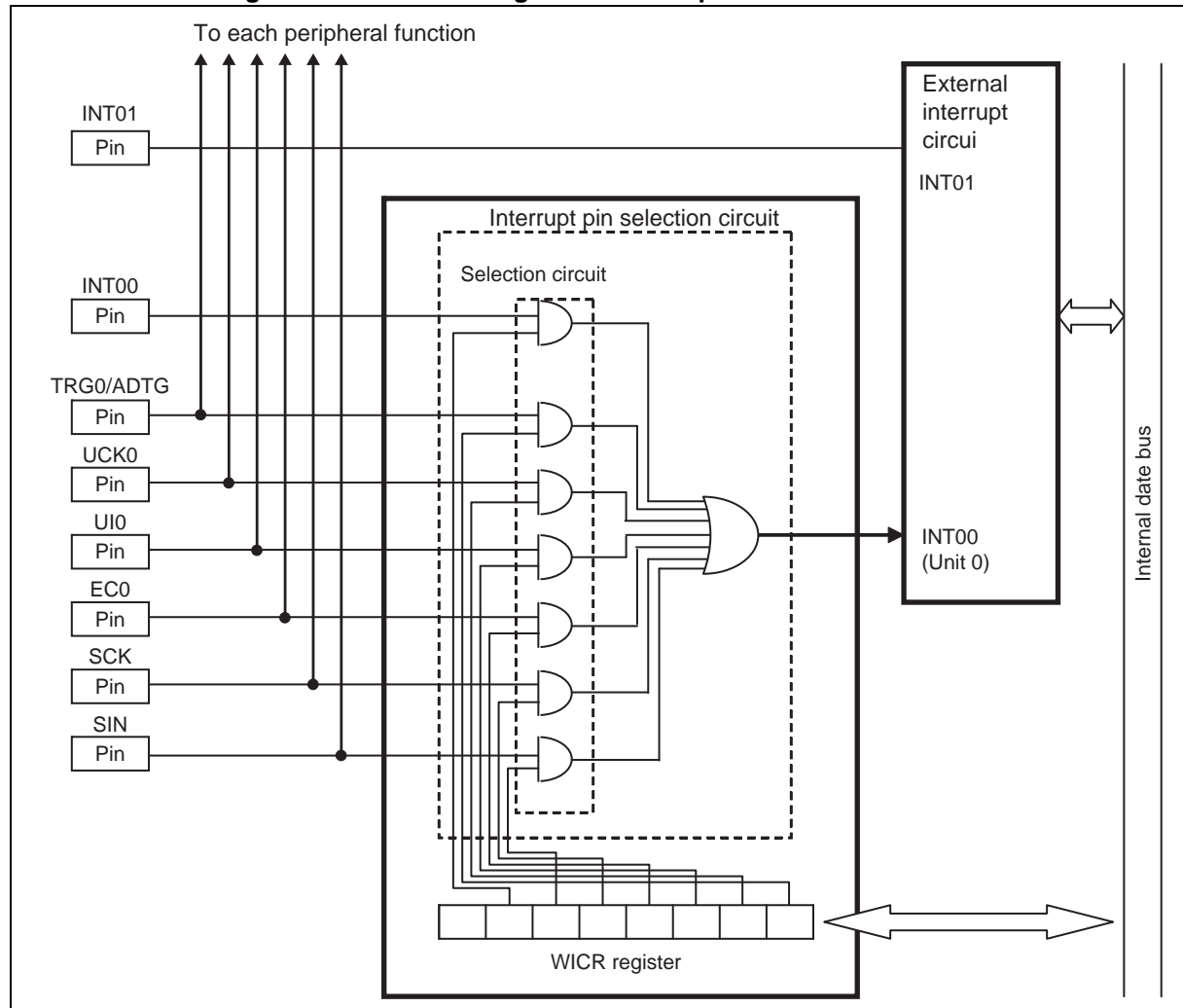
The interrupt pin selection circuit is used to select interrupt input pins from amongst various peripheral inputs (TRG0/ADTG, UCK0, UI0, EC0, SCK, SIN, INT00). The input signal from each peripheral function pin is selected by this circuit and the signal is used as the INT00 (channel 0) input of external interrupt. This enables the input signals to the peripheral function pins to also serve as external interrupt pins.

## 19.2 Configuration of Interrupt Pin Selection Circuit

Figure 19.2-1 shows the block diagram of the interrupt pin selection circuit.

### ■ Block Diagram of Interrupt Pin Selection Circuit

Figure 19.2-1 Block Diagram of Interrupt Pin Selection Circuit



● WICR register (interrupt pin selection circuit control register)

This register is used to determine which of the available peripheral input pins should be outputted to the interrupt circuit and which interrupt pins they should serve as.

● Selection circuit

This circuit outputs the input from the pin selected by the WICR register to the INT00 input of the external interrupt circuit (ch.0).

## 19.3 Pins of Interrupt Pin Selection Circuit

---

**This section describes the pins of the interrupt pin selection circuit.**

---

### ■ Pins Related to Interrupt Pin Selection Circuit

The peripheral function pins related to the interrupt pin selection circuit are the TRG0/ADTG, UCK0, UI0, EC0, SCK, SIN, and INT00 pins. These inputs (except INT00) are also connected to their respective peripheral units in parallel and can be used for both functions simultaneously. Table 19.3-1 lists the correlation between the peripheral functions and peripheral input pins.

**Table 19.3-1 Correlation Between Peripheral Functions and Peripheral Input Pins**

Peripheral input pin name	Peripheral functions name
INT00	Interrupt pin selection circuit
TRG0/ADTG	Interrupt pin selection circuit 16-bit PPG timer (trigger input) 8/10-bit A/D converter (trigger input)
UCK0	Interrupt pin selection circuit UART/SIO (clock input/output)
UI0	Interrupt pin selection circuit UART/SIO (data input)
EC0	Interrupt pin selection circuit 8/16-bit compound timer (event input)
SCK	Interrupt pin selection circuit LIN-UART (clock input/output)
SIN	Interrupt pin selection circuit LIN-UART (data input)

## 19.4 Registers of Interrupt Pin Selection Circuit

Figure 19.4-1 shows the registers related to the interrupt pin selection circuit.

### ■ Registers Related to Interrupt Pin Selection Circuit

**Figure 19.4-1 Registers Related to Interrupt Pin Selection Circuit**

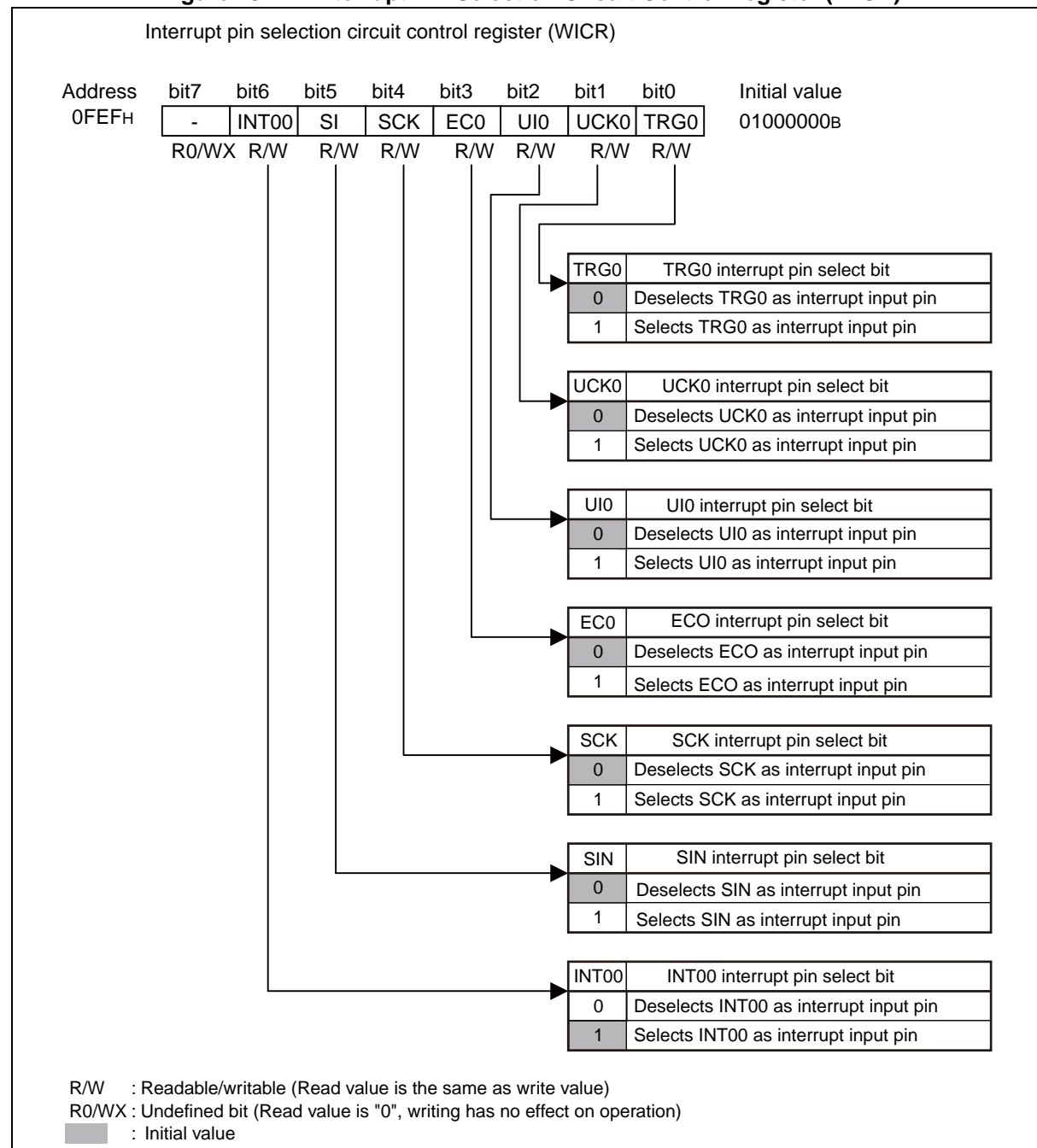
Interrupt pin selection circuit control register (WICR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FEF <sub>H</sub>	–	INT00	SIN	SCK	EC0	UI0	UCK0	TRG0
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
01000000 <sub>B</sub>								
R/W : Readable/writable (Read value is the same as write value)								
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)								

## 19.4.1 Interrupt Pin Selection Circuit Control Register (WICR)

This register is used to determine which of the available peripheral input pins should be outputted to the interrupt circuit and which interrupt pins they should serve as.

### ■ Interrupt Pin Selection Circuit Control Register (WICR)

Figure 19.4-2 Interrupt Pin Selection Circuit Control Register (WICR)



**Table 19.4-1 Functional Description of Each Bit of Interrupt Pin Selection Circuit Control Register (WICR) (1 / 2)**

Bit name		Function
bit7	Undefined bit	<p>This bit is undefined.</p> <ul style="list-style-type: none"> <li>• The read value is always "0".</li> <li>• Writing has no effect on the operation.</li> </ul>
bit6	INT00: IINT00 interrupt pin select bit	<p>This bit is used to determine whether to select the INT00 pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the INT00 pin as an interrupt input pin and the circuit treats the INT00 pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the INT00 pin as an interrupt input pin and the circuit passes the INT00 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the INT00 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>
bit5	SIN: SIN interrupt pin select bit	<p>This bit is used to determine whether to select the SIN pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the SIN pin as an interrupt input pin and the circuit treats the SIN pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the SIN pin as an interrupt input pin and the circuit passes the SIN pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SIN pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>
bit4	SCK: SCK interrupt pin select bit	<p>This bit is used to determine whether to select the SCK pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the SCK pin as an interrupt input pin and the circuit treats the SCK pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the SCK pin as an interrupt input pin and the circuit passes the SCK pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SCK pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>
bit3	EC0: EC0 interrupt pin select bits	<p>This bit is used to determine whether to select the EC0 pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the EC0 pin as an interrupt input pin and the circuit treats the EC0 pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the EC0 pin as an interrupt input pin and the circuit passes the EC0 pin input to INT000 (ch.0) of the external interrupt circuit. In this case, the input signal to the EC0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>
bit2	UI0: UI0 interrupt pin select bits	<p>This bit is used to determine whether to select the UI0 pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the UI0 pin as an interrupt input pin and the circuit treats the UI0 pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the UI0 pin as an interrupt input pin and the circuit passes the UI0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the UI0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>
bit1	UCK0: UCK0 interrupt pin select bit	<p>This bit is used to determine whether to select the UCK0 pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the UCK0 pin as an interrupt input pin and the circuit treats the UCK0 pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the UCK0 pin as an interrupt input pin and the circuit passes the UCK0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the UCK0 pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>

**Table 19.4-1 Functional Description of Each Bit of Interrupt Pin Selection Circuit Control Register (WICR) (2 / 2)**

Bit name		Function
bit0	TRG0: TRG0 interrupt pin select bit	<p>This bit is used to determine whether to select the TRG0 pin as an interrupt input pin.</p> <p><b>Setting the bit to "0":</b> Deselects the TRG0 pin as an interrupt input pin and the circuit treats the TRG0 pin input as being fixed at "0".</p> <p><b>Setting the bit to "1":</b> Selects the TRG0 pin as an interrupt input pin and the circuit passes the TRG0 pin input to INT00 (ch.0) of the external interrupt circuit. In this case, the input signal to the SCK pin can generate an external interrupt if INT00 (ch.0) operation is enabled in the external interrupt circuit.</p>

When these bits are set to "1" and the operation of INT00 (ch.0) of the external interrupt circuit is enabled in MCU standby mode, the selected pins are enabled to perform input operation. The MCU wakes up from the standby mode when a valid edge pulse is inputted to the pins. For information about the standby modes, refer to "6.8 Operations in Low-power Consumption Modes (Standby Modes)".

**Note:**

The input signals to the peripheral pins do not generate an external interrupt even when "1" is written to these bits if the INT00 (ch.0) of the external interrupt circuit is disabled.

Do not modify the values of these bits while the INT00 (ch.0) of the external interrupt circuit is enabled. If modified, the external interrupt circuit may detect a valid edge, depending on the pin input level.

If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch.0) of the external interrupt circuit is enabled (the values other than "00<sub>B</sub>" are set to SL01, SL00 bits in EIC00 register of external interrupt circuit and the interrupt is enabled by writing "1" to the EIE0 bit when selecting the valid edge), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.

## 19.5 Operating Description of Interrupt Pin Selection Circuit

---

**The interrupt pins are selected by setting WICR (interrupt pin selection circuit control register).**

---

### ■ Operation of Interrupt Pin Selection Circuit

The WICR (interrupt pin selection circuit control register) setting is used to select the input pins to be inputted to INT00 of the external interrupt circuit (ch.0). Shown below is the setup procedure for the interrupt pin selection circuit and external interrupt circuit (ch.0), which must be followed when selecting the TRG0 pin as an interrupt pin.

- 1) Write "0" to the corresponding bit in the port direction register (DDR) to set the pin as an input.
- 2) Select the TRG0 pin as an interrupt input pin in WICR (interrupt pin selection circuit control register).  
(Write "01<sub>H</sub>" to the WICR register. At this point, after writing "0" in the EIE0 bit of the EIC00 register of the external interrupt circuit, the operation of the external interrupt circuit is disabled).
- 3) Enable the operation of INT00 of the external interrupt circuit (ch.0).  
(Set the SL01 and SL00 bits of the EIC00 register to any value other than "00<sub>B</sub>" in the external interrupt circuit to select the valid edge. Also write "1" to the EIE0 bit to enable interrupts).
- 4) The subsequent interrupt operation is the same as for the external interrupt circuit.

When a reset is released, WICR (interrupt pin selection circuit control register) is initialized to "40<sub>H</sub>" and the INT00 bit is selected as the only available interrupt pin. Update the value of this register before enabling the operation of the external interrupt circuit, when using any pins other than the INT00 pin as external interrupt pins.

---

**Note:**

If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously, an input to INT00 (ch.0) of the external interrupt circuit is treated as "H" if any of the selected input signals is "H". (It becomes "OR" of the signals inputted to the selected pins.)

---



## **19.6 Notes on Using Interrupt Pin Selection Circuit**

---

**This section explains the precautions to be taken when using the interrupt pin selection circuit.**

---

### **■ Notes on Using Interrupt Pin Selection Circuit**

- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously and the operation of INT00 (ch.0) of the external interrupt circuit is enabled (the values other than "00<sub>B</sub>" are set to SL01, SL00 bits in EIC00 register of external interrupt circuit and the interrupt is enabled by writing "1" to the EIE0 bit when selecting the valid edge), the selected pins will remain enabled to perform input so as to accept interrupts even in a standby mode.
- If more than one interrupt pin are selected in WICR (interrupt pin selection circuit control register) simultaneously, an input to INT00 (ch.0) of the external interrupt circuit is treated as "H" level if any of the selected input signals is "H" level (it becomes "OR" of the signals inputted to the selected pins).

# **CHAPTER 20**

---

## **UART/SIO**

**This chapter describes the functions and operations of UART/SIO.**

- 20.1 Overview of UART/SIO
- 20.2 Configuration of UART/SIO
- 20.3 Channels of UART/SIO
- 20.4 Pins of UART/SIO
- 20.5 Registers of UART/SIO
- 20.6 Interrupts of UART/SIO
- 20.7 Explanation of UART/SIO Operations and Setup  
Procedure Example
- 20.8 Sample Programs for UART/SIO

## 20.1 Overview of UART/SIO

The UART/SIO is a general-purpose serial data communication interface. Serial data transfers of variable-length data can be made with a synchronous or asynchronous clock. The transfer format is NRZ. The transfer rate can be set with the dedicated baud rate generator or external clock (in clock synchronous mode).

### ■ Functions of UART/SIO

The UART/SIO is capable of serial data transmission/reception (serial input/output) to and from another CPU or peripheral device.

- Equipped with a full-duplex double buffer that allows 2-way full-duplex communication.
- The synchronous or asynchronous transfer mode can be selected.
- The optimum baud rate can be selected with the dedicated baud rate generator.
- The data length is variable; it can be set to 5 bits to 8 bits when no parity is used or to 6 bits to 9 bits when parity is used. (Refer to Table 20.1-1).
- The serial data direction (endian) can be selected.
- The data transfer format is NRZ (Non-Return-to-Zero).
- Two operation modes (operation modes 0 and 1) are available.  
Operation mode 0 operates as asynchronous clock mode (UART).  
Operation mode 1 operates as clock synchronous mode (SIO).

**Table 20.1-1 Operation Modes of UART/SIO**

Operation mode	Data length		Synchronous mode	Stop bit length
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	—	Synchronous	—
	6	—		
	7	—		
	8	—		

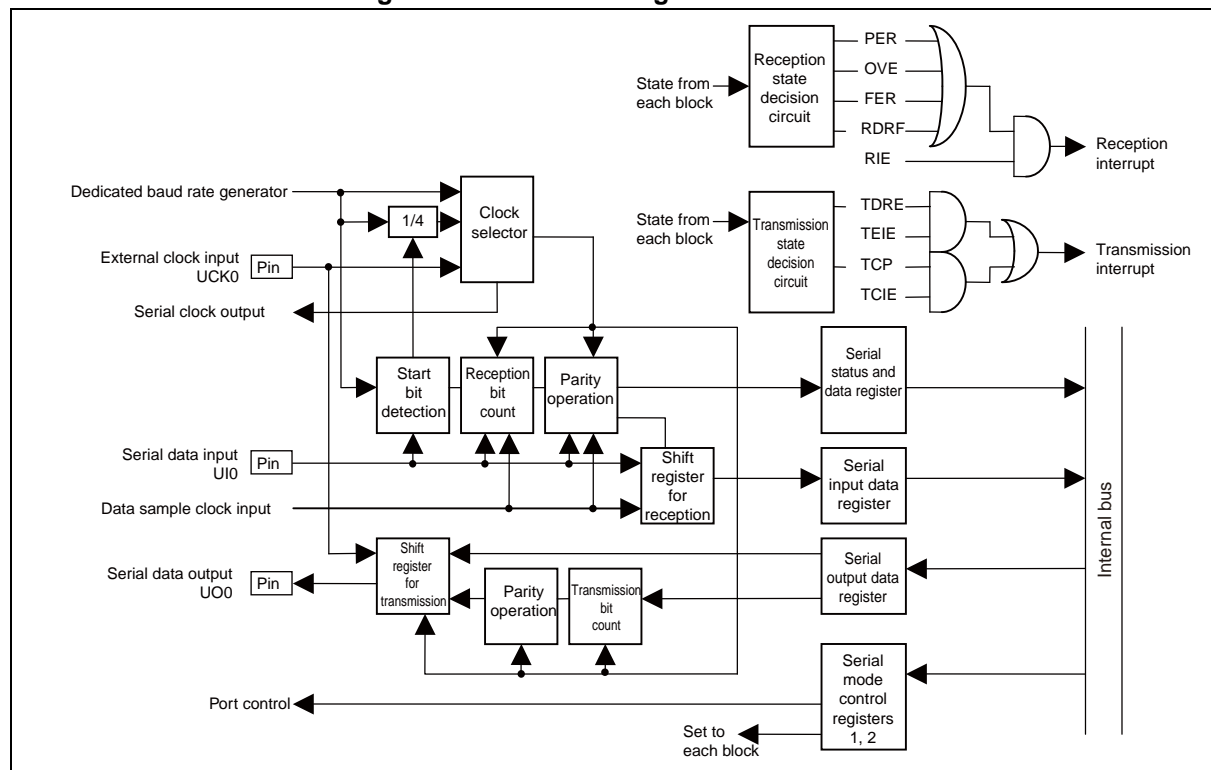
## 20.2 Configuration of UART/SIO

The UART/SIO consists of the following blocks:

- UART/SIO serial mode control register 1 (SMC10)
- UART/SIO serial mode control register 2 (SMC20)
- UART/SIO serial status and data register (SSR0)
- UART/SIO serial input data register (RDR0)
- UART/SIO serial output data register (TDR0)

### ■ Block Diagram of UART/SIO

Figure 20.2-1 Block Diagram of UART/SIO



- **UART/SIO serial mode control register 1 (SMC10)**

This register controls UART/SIO operation mode. The register is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/asynchronous), data length, and serial clock.

- **UART/SIO serial mode control register 2 (SMC20)**

This register controls UART/SIO operation mode. It is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

- **UART/SIO serial status and data register (SSR0)**

This register indicates the transmission/reception status and error status of UART/SIO.

- **UART/SIO serial input data register (RDR0)**

This register holds the receive data. The serial input is converted and then stored in this register.

- **UART/SIO serial output data register (TDR0)**

This register sets the transmit data. Data written to this register is serial-converted and then outputted.

## ■ **Input Clock**

The UART/SIO uses the output clock (internal clock) from the dedicated baud rate generator or the input signal (external clock) from the UCK0 pin as its input clock (serial clock).

## 20.3 Channels of UART/SIO

---

This section describes the channels of UART/SIO.

---

### ■ Channels of UART/SIO

This series contains one channel of the UART/SIO.

Table 20.3-1 and Table 20.3-2 show the correspondence of the channel, pin, and register.

**Table 20.3-1 Pins of UART/SIO**

Channel	Pin name	Pin function
0	UCK0	Clock input/output
	UO0	Data output
	UI0	Data input

**Table 20.3-2 Registers of UART/SIO**

Channel	Register name	Corresponding register (Representation in this manual)
0	SMC10	UART/SIO serial mode control register 1
	SMC20	UART/SIO serial mode control register 2
	SSR0	UART/SIO serial status and data register
	TDR0	UART/SIO serial output data register
	RDR0	UART/SIO serial input data register

## **20.4 Pins of UART/SIO**

---

**This section describes the pins related to the UART/SIO.**

---

### **■ Pins Related to UART/SIO**

The pins associated with UART/SIO are the clock input and output pin (UCK0), serial data output pin (UO0) and serial data input pin (UI0).

**UCK0:**

Clock input/output pin for UART/SIO.

When the clock output is enabled (SMC20:SCKE=1), it serves as a UART/SIO clock output pin (UCK0) regardless of the value of the corresponding port direction register. At this time, do not select the external clock (set SMC10:CKS = 0).

When it is to be used as a UART/SIO clock input pin, disable the clock output (SMC20:SCKE = 0) and make sure that it is set as input port by the corresponding port direction register. At this time, be sure to select the external clock (set SMC10:CKS = 0).

**UO0:**

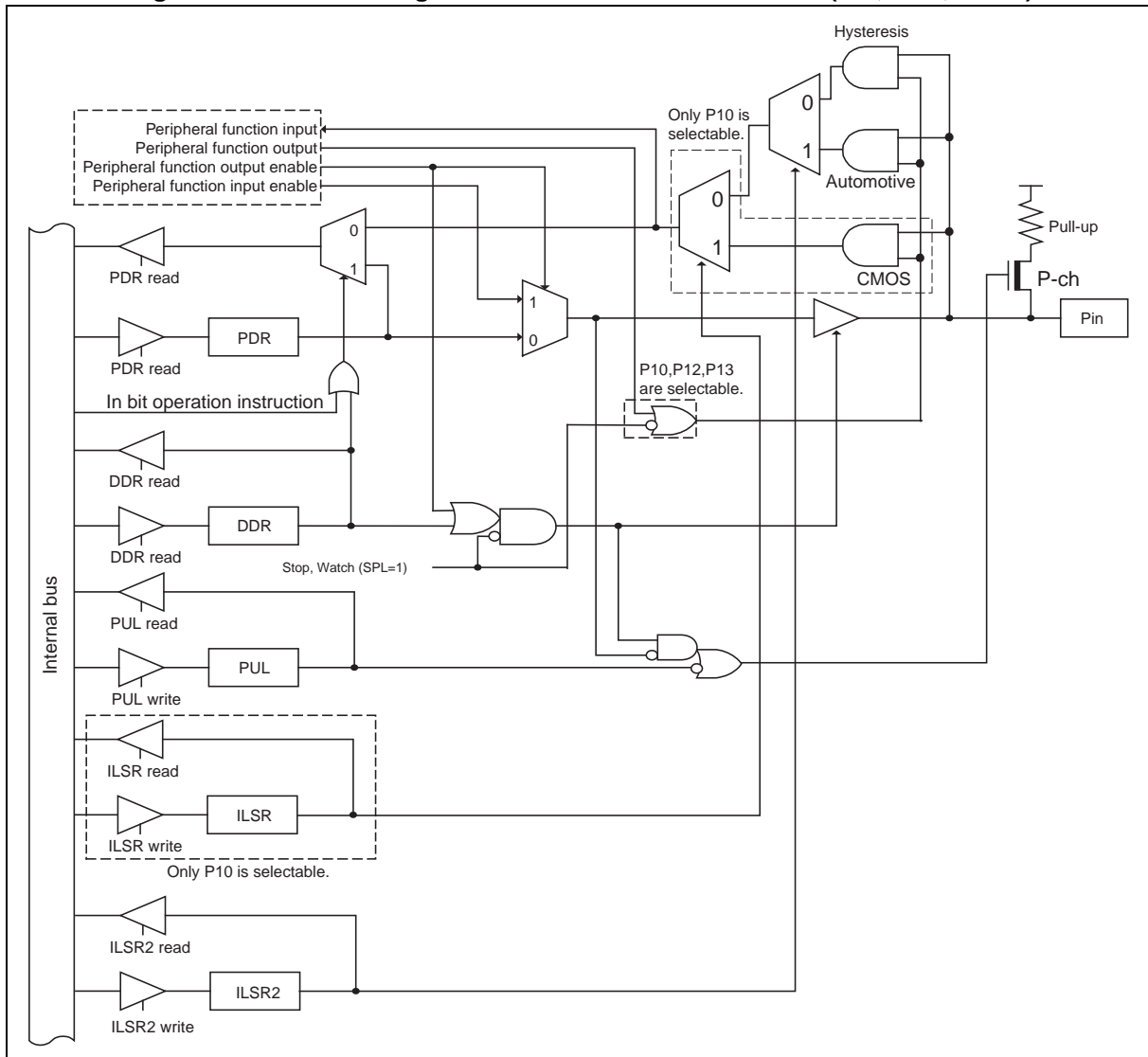
Serial data output pin for UART/SIO. When the serial data output is enabled (SMC20:TXOE = 1), it serves as a UART/SIO serial data output pin (UO0) regardless of the value of the corresponding port direction register.

**UI0:**

Serial data input pin for UART/SIO. When it is to be used as a UART/SIO serial data input pin, make sure that it is set as input port by the corresponding port direction register.

■ Block Diagram of Pins Related to UART/SIO

Figure 20.4-1 Block Diagram of Pins Related to UART/SIO (UI0, UO0, UCK0)





## 20.5 Registers of UART/SIO

The registers related to UART/SIO are UART/SIO serial mode control register 1 (SMC10), UART/SIO serial mode control register 2 (SMC20), UART/SIO serial status and data register (SSR0), UART/SIO serial output data register (TDR0), and UART/SIO serial input data register (RDR0).

### ■ Registers Related to UART/SIO

Figure 20.5-1 Registers Related to UART/SIO

UART/SIO serial mode control register 1 (SMC10)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0056 <sub>H</sub>	BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
UART/SIO serial mode control register 2 (SMC20)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0057 <sub>H</sub>	SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
	R/W	R/W	R1/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00100000 <sub>B</sub>								
UART/SIO serial status and data register (SSR0)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0058 <sub>H</sub>	—	—	PER	OVE	FER	RDRF	TCPL	TDRE
	R0/WX	R0/WX	R/WX	R/WX	R/WX	R/WX	R(RM1),W	R/WX
Initial value								
00000001 <sub>B</sub>								
UART/SIO serial output data register (TDR0)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0059 <sub>H</sub>	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
UART/SIO serial input data register (RDR0)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
005A <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX
Initial value								
00000000 <sub>B</sub>								
R/W : Readable/writable (Read value is the same as write value) R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction) R/WX : Read only (Readable, writing has no effect on operation) R0/WX : Undefined bit (Read value is "0", writing has no effect on operation) R1/W : Readable/writable (Read value is always "1")								

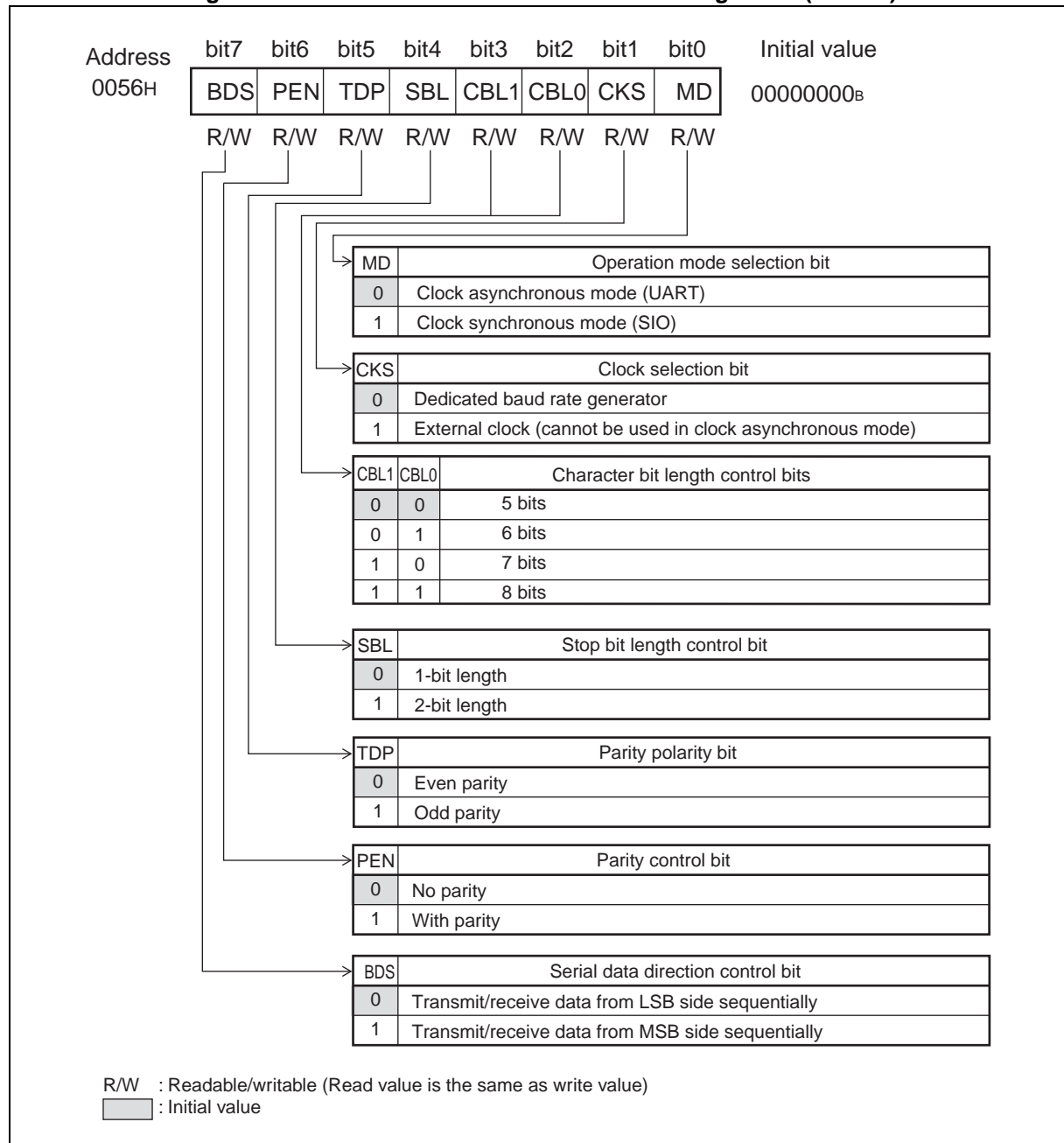
## MB95110B/M Series

### 20.5.1 UART/SIO Serial Mode Control Register 1 (SMC10)

UART/SIO serial mode control register 1(SMC10) controls the UART/SIO operation mode. The register is used to set the serial data direction (endian), parity and its polarity, stop bit length, operation mode (synchronous/asynchronous), data length, and serial clock.

#### ■ UART/SIO Serial Mode Control Register 1 (SMC10)

Figure 20.5-2 UART/SIO Serial Mode Control Register 1 (SMC10)



**Table 20.5-1 Functional Description of Each Bit of UART/SIO Serial Mode Control Register 1 (SMC10)**

Bit name		Function															
bit7	BDS: Serial data direction control bit	This bit sets the serial data direction (endian). <b>Setting the bit to "0"</b> : the bit specifies transmission or reception to be performed sequentially starting from the LSB side in the serial data register. <b>Setting the bit to "1"</b> : the bit specifies transmission or reception to be performed sequentially starting from the MSB side in the serial data register.															
bit6	PEN: Parity control bit	This bit enables or disables parity in clock asynchronous mode. <b>Setting the bit to "0"</b> : no parity <b>Setting the bit to "1"</b> : with parity															
bit5	TDP: Parity polarity bit	This bit controls even/odd parity. <b>Setting the bit to "0"</b> : specifies even parity <b>Setting the bit to "1"</b> : specifies odd parity															
bit4	SBL: Stop bit length control bit	This bit controls the length of the stop bit in clock asynchronous mode. <b>Setting the bit to "0"</b> : sets the stop bit length to "1". <b>Setting the bit to "1"</b> : sets the stop bit length to "2". Note: The setting of this bit is only valid for transmission operation in asynchronous mode. For receiving operation, reception data register full flag is set to "1" after detecting stop bit (1-bit) and completing the reception regardless of this bit.															
bit3, bit2	CBL1, CBL0: Character bit length control bit	These bits select the character bit length as shown in the following table: <table border="1"> <thead> <tr> <th>CBL1</th><th>CBL0</th><th>Character bit length</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>5</td></tr> <tr> <td>0</td><td>1</td><td>6</td></tr> <tr> <td>1</td><td>0</td><td>7</td></tr> <tr> <td>1</td><td>1</td><td>8</td></tr> </tbody> </table> <p>The above setting is valid in both asynchronous and synchronous modes.</p>	CBL1	CBL0	Character bit length	0	0	5	0	1	6	1	0	7	1	1	8
CBL1	CBL0	Character bit length															
0	0	5															
0	1	6															
1	0	7															
1	1	8															
bit1	CKS: Clock selection bit	This bit selects the external clock or dedicated baud rate generator. <b>Setting the bit to "0"</b> : selects the dedicated baud rate generator. <b>Setting the bit to "1"</b> : selects the external clock. Note: Setting this bit to "1" forcibly disables the output of the UCK0 pin. The external clock cannot be used in clock asynchronous mode (UART).															
bit0	MD: Operation mode selection bit	This bit selects clock asynchronous mode (UART) or clock synchronous mode (SIO). <b>Setting the bit to "0"</b> : selects clock asynchronous mode (UART). <b>Setting the bit to "1"</b> : selects clock synchronous mode (SIO).															

**Note:**

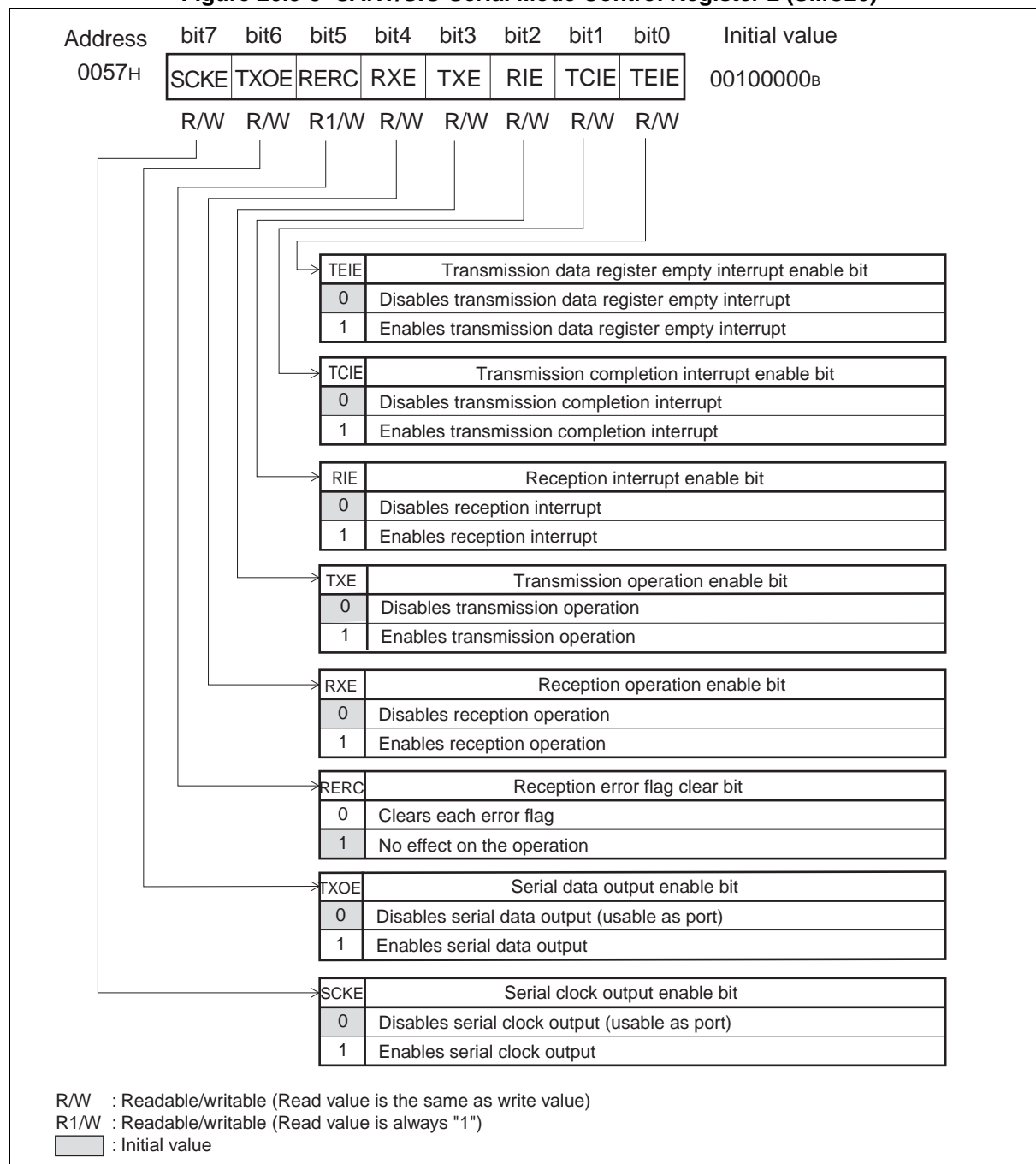
When modifying the UART/SIO serial mode control register 1 (SMC10), do not perform the modification during data transmission or reception.

## 20.5.2 UART/SIO Serial Mode Control Register 2 (SMC20)

UART/SIO serial mode control register 2 (SMC20) controls the UART/SIO operation mode. The register is used to enable/disable serial clock output, serial data output, transmission/reception, and interrupts and to clear the reception error flag.

## ■ UART/SIO Serial Mode Control Register 2 (SMC20)

Figure 20.5-3 UART/SIO Serial Mode Control Register 2 (SMC20)



**Table 20.5-2 Functional Description of Each Bit of UART/SIO Serial Mode Control Register 2 (SMC20)**

Bit name		Function
bit7	SCKE: Serial clock output enable bit	This bit controls the input/output of the serial clock (UCK0) pin in clock synchronous mode. <b>Setting the bit to "0"</b> : allows the pin to be used as a general-purpose port. <b>Setting the bit to "1"</b> : enables clock output. Note: When CKS is 1, the internal clock signal is not outputted even with this bit set to "1". If this bit is set to "1" with SMC10:MD set to "0" (asynchronous mode), the output from the port will always be "H".
bit6	TXOE: Serial data output enable bit	This bit controls the output of the serial data (UO0 pin). <b>Setting the bit to "0"</b> : allows the pin to be used as a general-purpose port. <b>Setting the bit to "1"</b> : enables serial data output.
bit5	RERC: Receive error flag clear bit	<b>Setting the bit to "0"</b> : clears the error flags (PER, OVE, FER) of the SSR0 register. <b>Setting the bit to "1"</b> : has no effect on operation. Reading this bit always returns "1".
bit4	RXE: Reception operation enable bit	<b>Setting the bit to "0"</b> : disables the reception of serial data. <b>Setting the bit to "1"</b> : enables the reception of serial data. If this bit is set to "0" during reception, the reception operation will be immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the serial input data register. Note: Setting this bit to "0" initializes reception operation. It has no effect on the receive data register full (RDRF) bit or an error flag (PER, OVE, FER).
bit3	TXE: Transmission operation enable bit	<b>Setting the bit to "0"</b> : disables the transmission of serial data. <b>Setting the bit to "1"</b> : enables the transmission of serial data. If this bit is set to "0" during transmission, the transmission operation will be immediately disabled and initialization will be performed. The transmission completion flag (TCPL) will be set to "1" and the transmission data register empty (TDRE) bit will also be set to "1".
bit2	RIE: Reception interrupt enable bit	<b>Setting the bit to "0"</b> : disables reception interrupt. <b>Setting the bit to "1"</b> : enables reception interrupt. A reception interrupt occurs immediately after either the receive data register full (RDRF) bit or an error flag (PER, OVE, FER) is set to "1" with this bit set to "1" (enabled).
bit1	TCIE: Transmission completion interrupt enable bit	<b>Setting the bit to "0"</b> : disables interrupts by the transmission completion flag. <b>Setting the bit to "1"</b> : enables interrupts by the transmission completion flag. A transmission interrupt occurs immediately after the transmission completion flag (TCPL) bit is set to "1" with this bit set to "1" (enabled).
bit0	TEIE: Transmission data register empty interrupt enable bit	<b>Setting the bit to "0"</b> : disables interrupts by the transmission data register empty. <b>Setting the bit to "1"</b> : enables interrupts by the transmission data register empty. A transmission interrupt occurs immediately after the transmission data register empty (TDRE) bit is set to "1" with this bit set to "1" (enabled).

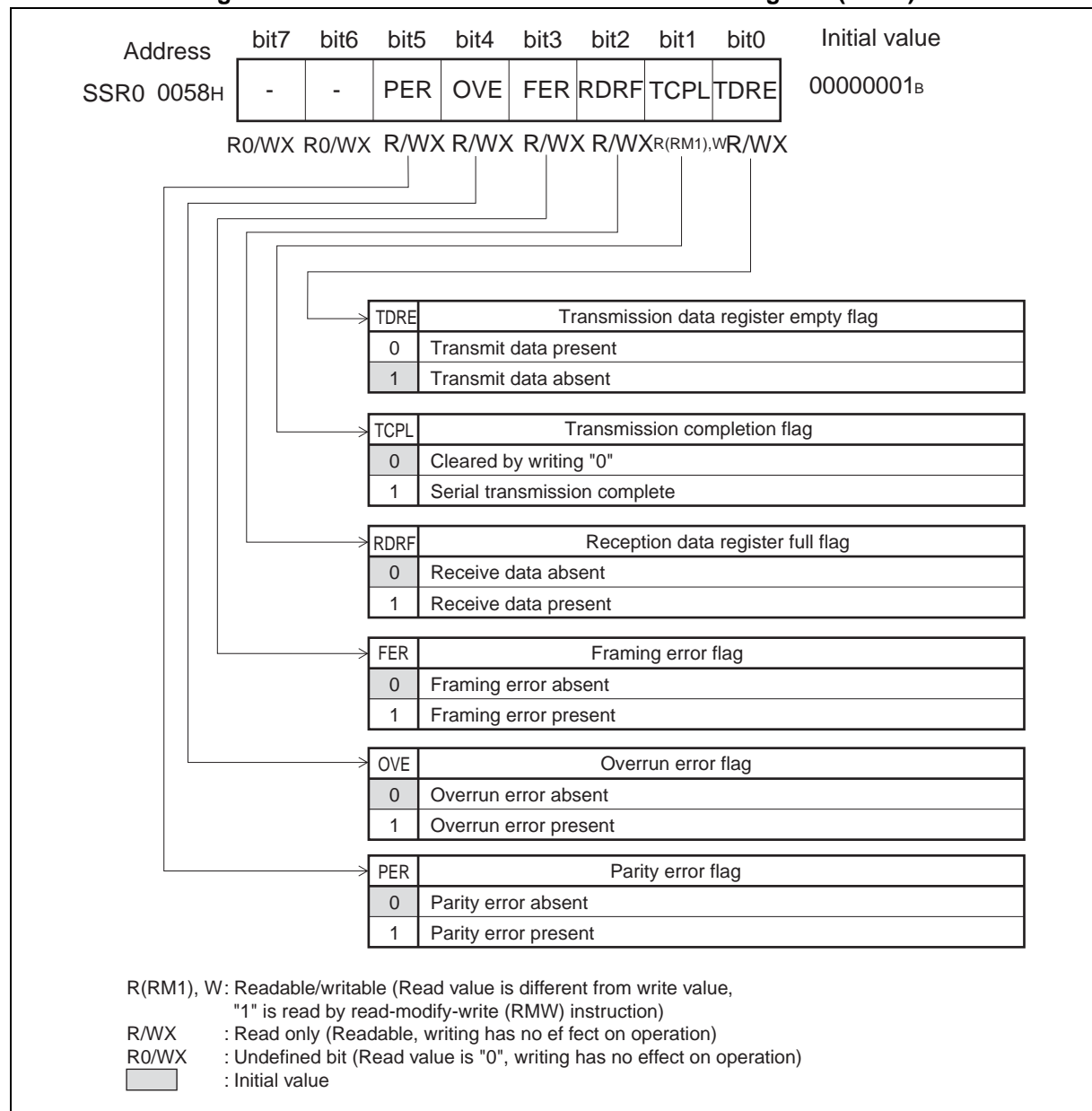
## MB95110B/M Series

### 20.5.3 UART/SIO Serial Status and Data Register (SSR0)

The UART/SIO serial status and data register (SSR0) indicates the transmission/reception status and error status of the UART/SIO.

#### ■ UART/SIO Serial Status and Data Register (SSR0)

Figure 20.5-4 UART/SIO Serial Status and Data Register (SSR0)



**Table 20.5-3 Functional Description of Each Bit of UART/SIO Serial Status and Data Register (SSR0)**

Bit name		Function
bit7, bit6	Undefined bits	<p>These bits are undefined.</p> <ul style="list-style-type: none"> <li>• Reading always returns "0".</li> <li>• Writing to the bits has no effect on operation.</li> </ul>
bit5	PER: Parity error flag	<p>Detect a parity error in received data.</p> <ul style="list-style-type: none"> <li>• The flag is set when a parity error occurs during reception. Writing "0" to the RERC bit clears this flag.</li> <li>• If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.</li> </ul>
bit4	OVE: Overrun error flag	<p>Detect an overrun error in received data.</p> <ul style="list-style-type: none"> <li>• The flag is set when an overrun error occurs during reception. Writing "0" to the RERC bit clears this flag.</li> <li>• If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.</li> </ul>
bit3	FER: Framing error flag	<p>Detect a framing error in received data.</p> <ul style="list-style-type: none"> <li>• The bit is set when a framing error occurs during reception. Writing "0" to the RERC bit clears this flag.</li> <li>• If error detection and clearing by RERC occur at the same time, the error flag is set preferentially.</li> </ul>
bit2	RDRF: Receive data register full flag	<p>This flag indicates the status of the UART/SIO serial input data register.</p> <ul style="list-style-type: none"> <li>• The bit is set to "1" when receive data is loaded to the serial input data register.</li> <li>• The bit is cleared to "0" when data is read from the serial input data register.</li> </ul>
bit1	TCPL: Transmission completion flag	<p>This flag indicates the data transmission status.</p> <ul style="list-style-type: none"> <li>• The bit is set to "1" upon completion of serial transmission. Note, however, that the bit is not set to "1" even upon completion of transmission when the serial output data register contains data to be transmitted in succession.</li> <li>• Writing "0" to this bit clears its flag.</li> <li>• If events to set and clear the flag occur at the same time, it is set preferentially.</li> <li>• Writing "1" to this bit has no effect on operation.</li> </ul>
bit0	TDRE: Transmission data register empty flag	<p>This flag indicates the status of the UART/SIO serial output data register.</p> <ul style="list-style-type: none"> <li>• The bit is set to "0" when transmit data is written to the serial output register.</li> <li>• The bit is set to "1" when data is loaded to the transmission shift register and transmission starts.</li> </ul>

## 20.5.4 UART/SIO Serial Input Data Register (RDR0)

The UART/SIO serial input data register (RDR0) is used to input (receive) serial data.

### ■ UART/SIO Serial Input Data Register (RDR0)

Figure 20.5-5 shows the bit configuration of the UART/SIO serial input data register (RDR0).

**Figure 20.5-5 UART/SIO Serial Input Data Register (RDR0)**

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
RDR0 005A <sub>H</sub>	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX: Read only (Readable, writing has no effect on operation)

This register stores received data. The serial data signals sent to the serial data input pin (UI0 pin) is converted by the shift register and stored in this register.

When received data is set correctly in this register, the receive data register full (RDRF) bit is set to "1". At this time, an interrupt occurs if reception interrupt requests have been enabled. If an RDRF bit check by the program or using an interruption shows that received data is stored in this register, the reading of the content for this register clears the RDRF flag to "0".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are set to "0".



## 20.5.5 UART/SIO Serial Output Data Register (TDR0)

The UART/SIO serial output data register (TDR0) is used to output (transmit) serial data.

### ■ UART/SIO Serial Output Data Register (TDR0)

Figure 20.5-6 shows the bit configuration of the UART/SIO serial output data register (TDR0).

**Figure 20.5-6 UART/SIO Serial Output Data Register (TDR0)**

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TDR0 0059 <sub>H</sub>	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

This register holds data to be transmitted. The register accepts a write when the transmission data register empty (TDRE) bit contains "1". An attempt to write to the bit is ignored when the bit contains "0".

When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of serial mode control register 2 is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible. Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR0, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". The transmission data is transferred to the shift register for the transmission, it is converted into the serial data, and it is transmitted from the serial data output pin.

When transmit data is written to the UART/SIO serial output data register (TDR0), the transmission data register empty bit (TDRE) is set to "0". Upon completion of transfer of transmit data to the transmission shift register, the transmission data register empty bit (TDRE) is set to "1", allowing the next piece of transmit data to be written. At this time, an interrupt occurs if transmission data register empty interrupts have been enabled. Write the next piece of transmit data when transmit data register empty occurs or the transmit data register empty (TDRE) bit is set to "1".

When the character bit length (CBL1, CBL0) is set to shorter than 8 bits, the excess upper bits (beyond the set bit length) are ignored.

#### Note:

The data in this register cannot be updated when TDRE in UART/SIO serial status and data register is "0".

When this register is updated at writing complete the transmission data and TDRE=0 (without depending on TXE of serial mode control register 2 is "1" or "0"), the transmission operation is initialized by writing "0" to TXE, TDRE becomes "1", and the update of this register becomes possible. Moreover, when "0" is written in TXE without the starting transmission (when the transmission data is written in TDR0, and it has not transmitted TXE to "1" yet), TCPL is not set in "1". And, to change data, please write it after making TDRE "1" once by writing TXE =0.

## 20.6 Interrupts of UART/SIO

The UART/SIO has six interrupt-related bits: error flag bits (PER, OVE, FER), receive data register full bit (RDRF), transmission data register empty bit (TDRE), and transmission completion flag (TCPL).

### ■ Interrupts of UART/SIO

Table 20.6-1 lists the UART/SIO interrupt control bits and interrupt sources.

**Table 20.6-1 UART/SIO Interrupt Control Bits and Interrupt Sources**

Item	Description					
Interrupt request flag bit	SSR0: TDRE	SSR0: TCPL	SSR0: RDRE	SSR0: PER	SSR0: OVE	SSR0: FER
Interrupt request enable bit	SMC20: TEIE	SMC20: TCIE	SMC20: RIE	SMC20: RIE	SMC20: RIE	SMC20: RIE
Interrupt source	Transmission data register empty	Transmission completion	Reception data register full	Parity error	Overrun error	Framing error

### ■ Transmit Interrupts

When transmit data is written to the serial output data register (TDR0), the data is transferred to the transmission shift register. When the next piece of data can be written, the TDRE bit is set to "1". At this time, an interrupt request to the interrupt controller occurs when transmit data register empty interrupt enable bit has been enabled (SMC20:TEIE = 1). The TCPL bit is set to "1" upon completion of transmission of all pieces of transmit data. At this time, an interrupt request to the interrupt controller occurs when transmission completion interrupt enable bit has been enabled (SMC20:TCIE = 1).

### ■ Reception Interrupt

If the data is inputted successfully up to the stop bit, the RDRF bit is set to 1. If an overrun, parity, or framing error occurs, the corresponding error flag bit (PER, OVE, or FER) is set to "1".

These bits are set when a stop bit is detected. If reception interrupt enable bit has been enabled (SMC20:RIE = 1), an interrupt request to the interrupt controller will be generated.

Refer to "CHAPTER 8 INTERRUPTS" for the interrupt request numbers and vector tables of all peripheral functions.

### ■ Registers and Vector Table Related to UART/SIO Interrupts

**Table 20.6-2 Registers and Vector Table Related to UART/SIO Interrupts**

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
ch.0	IRQ4	ILR1	L04	FFF2 <sub>H</sub>	FFF3 <sub>H</sub>

## Example

## 20.7 Explanation of UART/SIO Operations and Setup Procedure Example

---

The UART/SIO has a serial communication function (operation modes 0, 1).

---

### ■ Operation of UART/SIO

#### ● Operation mode

Two operation modes are available in the UART/SIO. Clock synchronous mode (SIO) or clock asynchronous mode (UART) can be selected (see Table 20.7-1).

**Table 20.7-1 Operation Modes of UART/SIO**

Operation mode	Data length		Synchronous Mode	Stop bit length
	No parity	With parity		
0	5	6	Asynchronous	1 bit or 2 bits
	6	7		
	7	8		
	8	9		
1	5	—	Synchronous	—
	6	—		
	7	—		
	8	—		

### ■ Setup Procedure Example

The UART/SIO is set up in the following procedure.

#### ● Initial setting

- 1) Set the port for input. (DDR1)
- 2) Set the interrupt level. (ILR1)
- 3) Set the prescaler. (PSSR0)
- 4) Set the baud rate. (BRSR0)
- 5) Select the clock. (SMC10:CKS)
- 6) Set the operation mode. (SMC10:MD)
- 7) Enable/disable the serial clock output. (SMC20:SCKE)
- 8) Enable reception. (SMC20:RXE = 1)
- 9) Enable interrupts. (SMC20:RIE = 1)

#### ● Interrupt processing

Read receive data. (RDR0)

## 20.7.1 Operating Description of Operation Mode 0

Operation mode 0 operates as clock asynchronous mode (UART).

### ■ Operating Description of UART/SIO Operation Mode 0

Clock asynchronous mode (UART) is selected when the MD bit in the UART/SIO serial mode control register 1 (SMC10) is set to "0".

#### ● Baud rate

The serial clock is selected by the CKS bit in the SMC10 register. Be sure to select the dedicated baud rate generator at this time.

The baud rate is equivalent to the output clock frequency of the dedicated baud rate generator, divided by four. The UART can perform communication within the range from -2% to +2% of the selected baud rate.

The baud rate generated by the dedicated baud rate generator is obtained from the equation illustrated below. (For information about the dedicated baud rate generator, refer to "CHAPTER 21 UART/SIO DEDICATED BAUD RATE GENERATOR".

**Figure 20.7-1 Baud Rate Calculation when Using Dedicated Baud Rate Generator**

$$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{4 \times \begin{matrix} 1 \\ 2 \\ 4 \\ 8 \end{matrix} \times \begin{matrix} 2 \\ : \\ 255 \end{matrix}} \quad [\text{bps}]$$

UART prescaler selection register (PSSR0)  
Prescaler selection (PSS1, PSS0)

UART baud rate setting register (BRSR0)  
Baud rate setting (BRS7 to BRS0)

**Table 20.7-2 Sample Asynchronous Transfer Rates Based on Dedicated Baud Rate Generator (Machine Clock = 10MHz, 16MHz, 16.25MHz)**

Dedicated baud rate generator setting		UART Internal division	Total division ratio (PSS × BRS × 4)	Baud rate (10MHz/ Total division ratio)	Baud rate (16MHz/ Total division ratio)	Baud rate (16.25MHz/ Total division ratio)
Prescaler selection PSS[1:0]	Baud rate counter setting BRS[7:0]					
1 (Setting value:0, 0)	20	4	80	125000	200000	203125
1 (Setting value:0, 0)	22	4	88	113636	181818	184659
1 (Setting value:0, 0)	44	4	176	56818	90909	92330
1 (Setting value:0, 0)	87	4	348	28736	45977	46695
1 (Setting value:0, 0)	130	4	520	19231	30769	31250
2 (Setting value:0, 1)	130	4	1040	9615	15385	15625
4 (Setting value:1, 0)	130	4	2080	4808	7692	7813
8 (Setting value:1, 1)	130	4	4160	2404	3846	3906

## Example

The baud rate in clock asynchronous mode can be set in the following range.

**Table 20.7-3 Baud Rate Setting Range in Clock Asynchronous Mode**

PSS[1:0]	BRS[7:0]
00 <sub>B</sub> to 11 <sub>B</sub>	02 <sub>H</sub> (2) to FF <sub>H</sub> (255)

● Transfer data format

UART can treat data only in NRZ (Non-Return-to-Zero) format. Figure 20.7-2 shows the transfer data format.

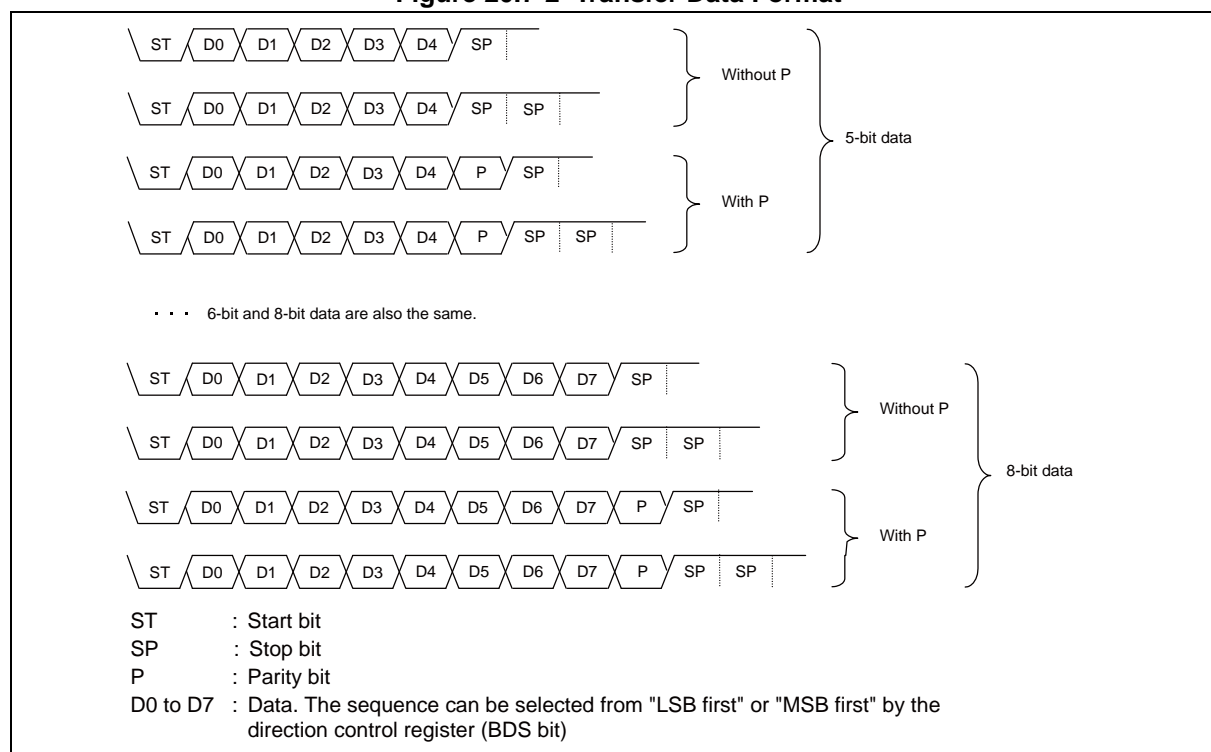
The character bit length can be selected from among 5 to 8 bits depending on the CBL1 and CBL0 settings.

The stop bit length can be set to 1 or 2 bits depending on the SBL setting.

PEN and TDP can be used to enable/disable parity and to select parity polarity.

As is shown in Figure 20.7-2, the transfer data always starts from the start bit ("L" level) and ends with the stop bit ("H" level) by performing the specified data bit length transfer with MSB first or LSB first ("LSB first" or "MSB first" can be selected by the BDS bit). It becomes "H" level at the idle state.

**Figure 20.7-2 Transfer Data Format**



● Receiving operation in asynchronous clock mode (UART)

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

Reception remains performed as long as the reception operation enable bit (RXE) contains "1".

Upon detection of a start bit in receive data with the reception operation enable bit (RXE) set to "1", one frame of data is received according to the data format set in UART/SIO serial control register 1 (SMC10).

When the reception of one frame of data has been completed, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next frame of serial data can be received.

When the UART/SIO serial input data register (RDR0) stores data, the receive data register full (RDRF) bit is set to "1".

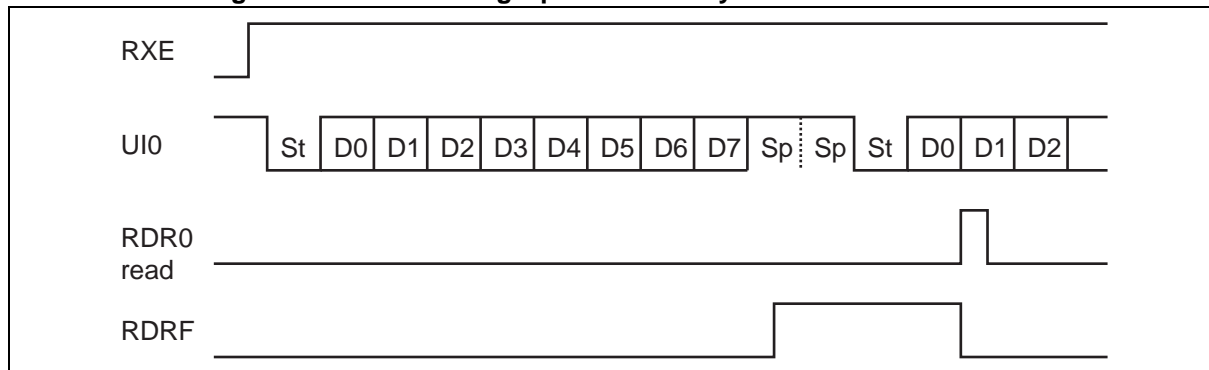
A reception interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the reception interrupt enable bit (RIE) contains "1".

Received data is read from the UART/SIO serial input data register (RDR0) after each error flag (PER, OVE, FER) in the UART/SIO serial status and data register is checked.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

Note that modifying UART/SIO serial mode control register 1 (SMC10) during reception may result in unpredictable operation. If the RXE bit is set to "0" during reception, the reception is immediately disabled and initialization will be performed. The data received up to that point will not be transferred to the serial input data register.

**Figure 20.7-3 Receiving Operation in Asynchronous Clock Mode**



## Example

## ● Reception error in asynchronous clock mode (UART)

If any of the following three error flags (PER, FER, OVE) has been set, receive data is not transferred to the UART/SIO serial input data register (RDR0) and the receive data register full (RDRF) bit is not set to "1" either.

- Parity error (PER)

The parity error (PER) bit is set to "1" if the parity bit in received serial data does not match the parity polarity bit (TDP) when the parity control bit (PEN) contains "1".

- Framing error (FER)

The framing error (FER) bit is set to "1" if "1" is not detected at the position of the first stop bit in serial data received in the set character bit length (CBL) under parity control (PEN).

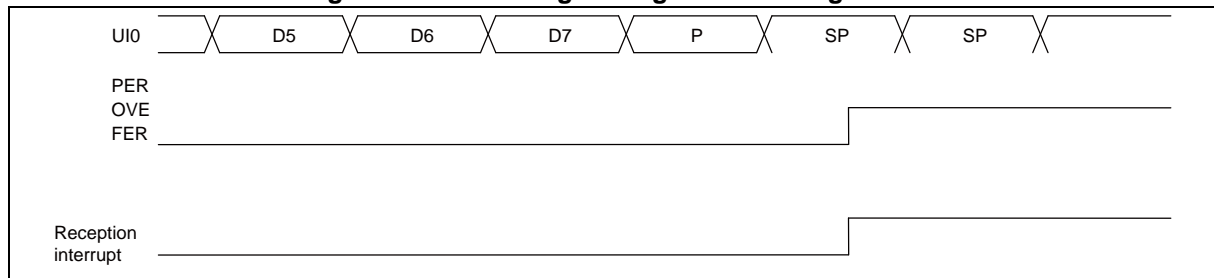
Note that the stop bit is not checked if it appears at the second bit or later.

- Overrun error (OVE)

Upon completion of reception of serial data, the overrun error (OVE) bit is set to "1" if the reception of the next data is performed before the previous receive data is read.

Each flag is set at the position of the first stop bit.

**Figure 20.7-4 Setting Timing for Receiving Errors**



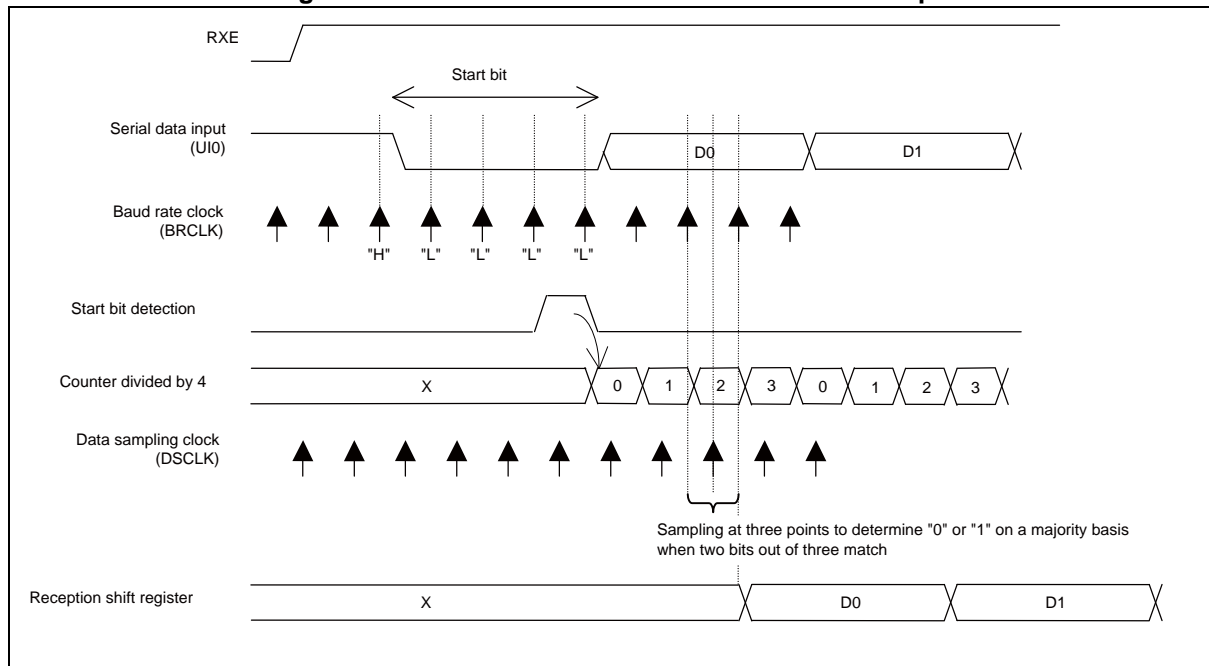
● Start bit detection and confirmation of receive data during reception

The start bit is detected by a falling of the serial input followed by a succession of three "L" levels after the serial data input is sampled according to the clock (BRCLK) signal provided by the dedicated baud rate generator with the reception operation enable bit (RXE) set to "1". When the first "H", "L", "L", "L", "L" train is detected in a BRCLK sample, therefore, the current bit is regarded as the start bit.

The frequency-quartered circuit is activated upon detection of the start bit and serial data is inputted to the reception shift register at intervals of four periods of BRCLK.

When data is received, sampling is performed at three points of the baud rate clock (BRCLK) and data sampling clock (DSCLK) and received data is confirmed on a majority basis when two bits out of three match.

**Figure 20.7-5 Start Bit Detection and Serial Data Input**





## Example

## ● Transmission in asynchronous clock mode

Use UART/SIO serial mode control register 1 (SMC10) to select the serial data direction (endian), parity/non-parity, parity polarity, stop bit length, character bit length, and clock.

The following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", and then write transmit data to the serial output data register to start transmission.
- Write transmit data to the serial output data register, and then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit is set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

The transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, and the transmit data register empty (TDRE) is set to "1".

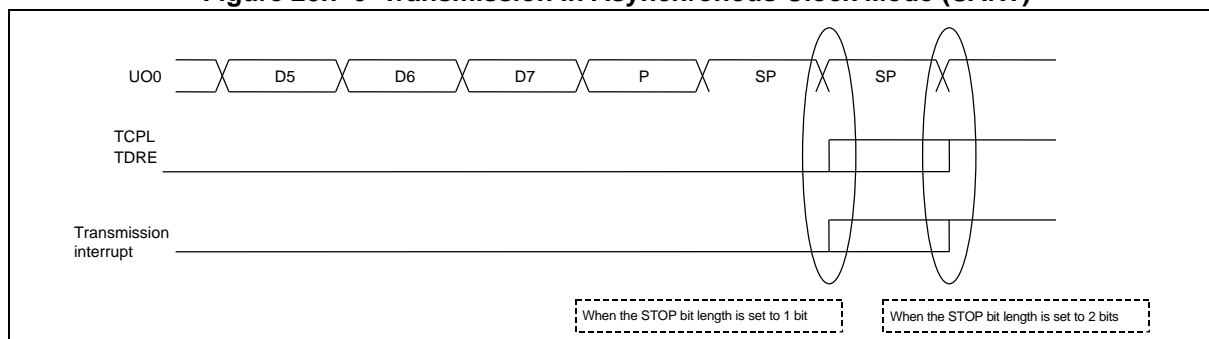
When the transmission interrupt enable bit (TIE) contains "1", a transmission interrupt occurs if the transmit data register empty (TDRE) bit is set to "1". This allows the next piece of transmit data to be written to the UART/SIO serial output data register (TDR0) by interrupt handling.

To detect the completion of serial transmission by transmission interrupt, set the transmission completion interrupt enable bits as follows: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmission interrupt occurs.

Both the transmission completion flag (TCPL) and the transmission data register empty flag (TDRE), when transmitting data consecutively, are set at the position which the transmission of the last bit was completed (it varies depending on the data length, parity enable, or stop bit length setting), as shown in Figure 20.7-6 below.

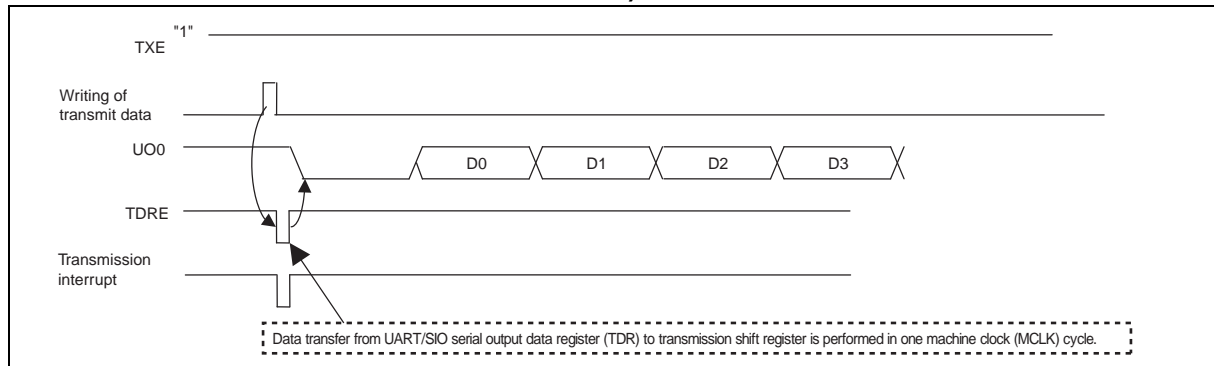
Note that modifying UART/SIO serial mode control register 1 (SMC10) during transmission may result in unpredictable operation.

**Figure 20.7-6 Transmission in Asynchronous Clock Mode (UART)**

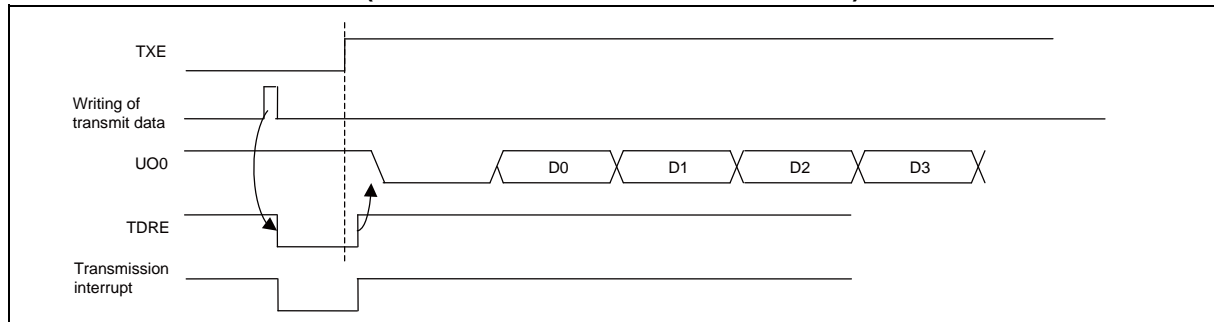


The TDRE flag is set at the point indicated in the following figure if the preceding piece of transmit data does not exist in the transmission shift register.

**Figure 20.7-7 Setting Timing 1 for Transmit Data Register Empty Flag (TDRE) (When TXE is "1")**



**Figure 20.7-8 Setting Timing 2 for Transmit Data Register Empty Flag (TDRE) (When TXE Is Switched from "0" to "1")**



### ● Concurrent transmission and reception

In asynchronous clock mode (UART), transmission and reception can be performed independently. Therefore, transmission and reception can be performed at the same time or even with transmitting and receiving frames overlapping each other in shifted phases.

## 20.7.2 Operating Description of Operation Mode 1

Operation mode 1 operates in synchronous clock mode.

### ■ Operating Description of UART/SIO Operation Mode 1

Setting the MD bit in UART/SIO serial mode control register 1 (SMC10) to "1" selects synchronous clock mode (SIO).

The character bit length in synchronous clock mode (SIO) is variable between 5 bits and 8 bits. Note, however, that parity is disabled and no stop bit is used.

The serial clock is selected by the CKS bit in the SMC10 register. Select the dedicated baud rate generator or external clock. The SIO performs shift operation using the selected serial clock as a shift clock.

To input the external clock signal, set the SCKE bit to "0".

To output the dedicated baud rate generator output as a shift clock signal, set the SCKE bit to "1". The serial clock signal is obtained by dividing clock by two, which is supplied by the dedicated baud rate generator. The baud rate in the SIO mode can be set in the following range. (For more information about the dedicated baud rate generator, also refer to "CHAPTER 21 UART/SIO DEDICATED BAUD RATE GENERATOR").

**Table 20.7-4 Baud Rate Setting Range in SIO Mode**

PSS[1:0]	BRS[7:0]
00 <sub>B</sub> to 11 <sub>B</sub>	01 <sub>H</sub> (1) to FF <sub>H</sub> (255), 00 <sub>H</sub> (256) (The highest and lowest baud rate settings are 01 <sub>H</sub> and 00 <sub>H</sub> , respectively.)

The baud rate applied when the external clock or dedicated baud rate generator is used is obtained from the corresponding equation illustrated below. (Figure 20.7-9, Figure 20.7-10)

**Figure 20.7-9 Calculating Baud Rate Based on External Clock**

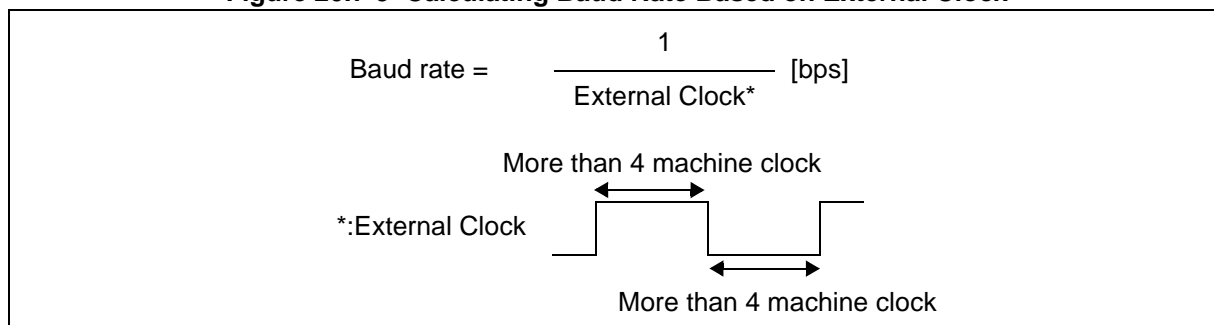


Figure 20.7-10 Baud Rate Calculation Formula for Using Dedicated Baud Rate Generator

$$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{2 \times \begin{matrix} 1 \\ 2 \\ 4 \\ 8 \end{matrix} \times \begin{matrix} 1 \\ : \\ 256 \end{matrix}} \quad [\text{bps}]$$

UART prescaler selection register (PSSR0)  
Prescaler selection (PSS1, PSS0)

UART baud rate setting register (BRSR0)  
Baud Rate Setting (BRS7 to BRS0)

● Serial clock

The serial clock signal is outputted under control of the output for transmit data. When only reception is performed, therefore, set transmission control (TXE = 1) to write dummy transmit data to the UART/SIO serial output register. Refer to the data sheet for the UCK0 clock value.

● Reception in UART/SIO operation mode 1

For reception in operation mode 1, each register is used as follows.

Figure 20.7-11 Registers Used for Reception in Operation Mode 1

SMC10 (UART/SIO Serial Mode Control Register 1)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
⊙	X	X	X	⊙	⊙	⊙	1
SMC20 (UART/SIO Serial Mode Control Register 2)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
⊙	0	⊙	⊙	⊙	⊙	X	X
SSR0 (UART/SIO serial status and data register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PER	OVE	FER	RDRF	TCPL	TDRE
X	X	X	⊙	X	⊙	X	X
TDR0 (UART/SIO serial output data register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
X	X	X	X	X	X	X	X
RDR0 (UART/SIO serial input data register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
⊙: Used bit X: Unused bit 0: Set "0" 1: Set "1"							

## Example

The reception depends on whether the serial clock has been set to external or internal clock.

**<When external clock is enabled>**

When the reception operation enable bit (RXE) contains "1", serial data is received always at the rising edge of the external clock signal.

**<When internal clock is enabled>**

The serial clock signal is outputted in accordance with transmission. Therefore, transmission must be performed even when only performing reception. The following two procedures can be used.

- Set the transmission operation enable bit (TXE) to "1", then write transmit data to the UART/SIO serial output data register to generate the serial clock signal and start reception.
- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to generate the serial clock signal and start reception.

When 5-bit to 8-bit serial data is received by the reception shift register, the received data is transferred to the UART/SIO serial input data register (RDR0) and the next piece of serial data can be received.

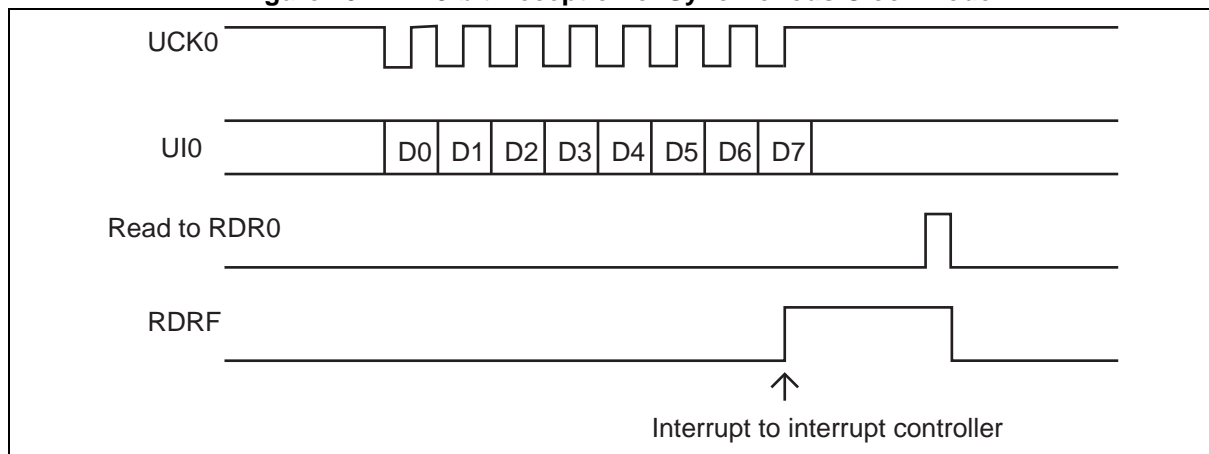
When the UART/SIO serial input data register stores data, the receive data register full (RDRF) bit is set to "1".

A reception interrupt occurs the moment the receive data register full (RDRF) bit is set to "1" when the reception interrupt enable bit (RIE) contains "1".

To read received data, read it from the UART/SIO serial input data register after checking the error flag (OVE) in the UART/SIO serial status and data register.

When received data is read from the UART/SIO serial input data register (RDR0), the receive data register full (RDRF) bit is cleared to "0".

**Figure 20.7-12 8-bit Reception of Synchronous Clock Mode**

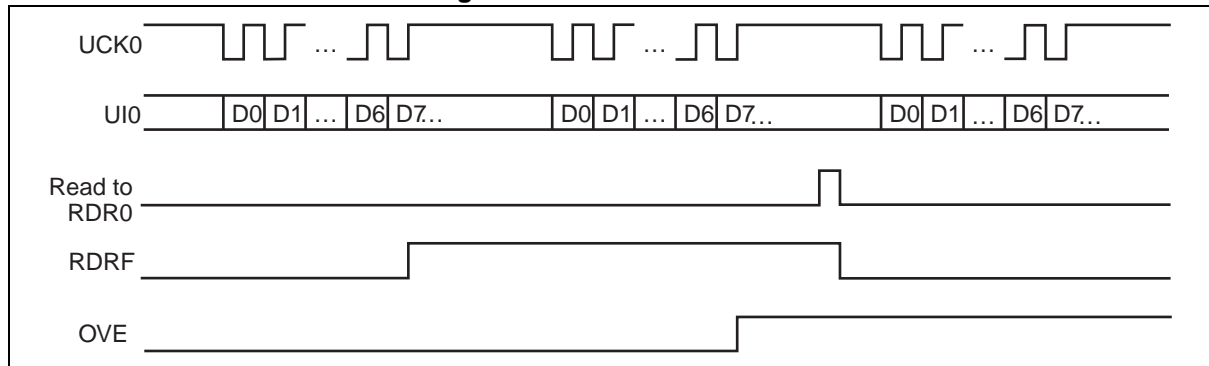
**Operation when reception error occurs**

When an overrun error (OVE) exists, received data is not transferred to the UART/SIO serial input data register (RDR0).

**Overrun error (OVE)**

Upon completion of reception for serial data, the overrun error (OVE) bit is set to "1" if the receive data register full (RDRF) bit has been set to "1" by the reception for the preceding piece of data.

Figure 20.7-13 Overrun error



● Transmission in UART/SIO operation mode 1

For transmission in operation mode 1, each register is used as follows.

Figure 20.7-14 Registers Used for Transmission in Operation Mode 1

SMC10 (UART/SIO Serial Mode Control Register 1)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
BDS	PEN	TDP	SBL	CBL1	CBL0	CKS	MD
⊙	X	X	X	⊙	⊙	⊙	1
SMC20 (UART/SIO Serial Mode Control Register 2)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCKE	TXOE	RERC	RXE	TXE	RIE	TCIE	TEIE
⊙	0	⊙	⊙	⊙	⊙	X	X
SSR0 (UART/SIO Serial Status and Data Register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
—	—	PER	OVE	FER	RDRF	TCPL	TDRE
X	X	X	⊙	X	⊙	X	X
TDR0 (UART/SIO Serial Output Data Register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
X	X	X	X	X	X	X	X
RDR0 (UART/SIO Serial Input Data Register)							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙

⊙: Used bit  
X: Unused bit  
0 : Set "0"  
1 :Set "1"

The following two procedures can be used to initiate the transmission process:

- Set the transmission operation enable bit (TXE) to "1", and then write transmit data to the UART/SIO serial output data register to start transmission.

**Example**

- Write transmit data to the UART/SIO serial output data register, then set the transmission operation enable bit (TXE) to "1" to start transmission.

Transmit data is written to the UART/SIO serial output data register (TDR0) after it is checked that the transmit data register empty (TDRE) bit is set to "1".

When the transmit data is written to the UART/SIO serial output data register (TDR0), the transmit data register empty (TDRE) bit is cleared to "0".

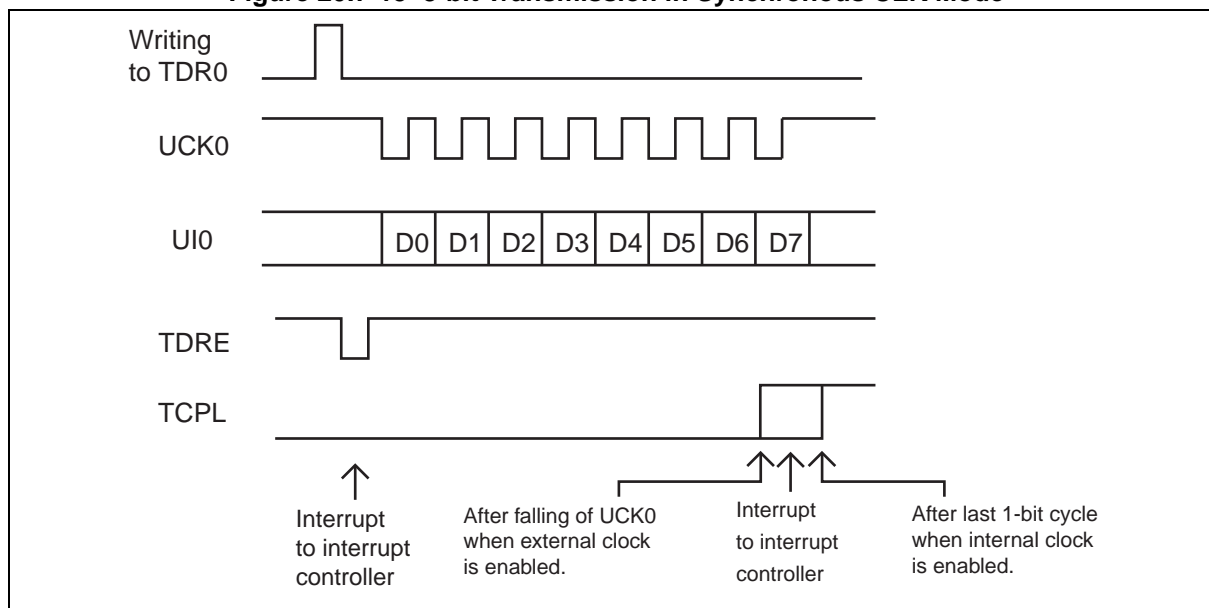
When serial transmission is started after transmit data is transferred from the UART/SIO serial output data register (TDR0) to the transmission shift register, and the transmit data register empty (TDRE) is set to "1".

When the use of the external clock signal has been set, serial data transmission starts at the fall of the first serial clock signal after the transmission process is started.

A transmission completion interrupt occurs the moment the transmit data register empty (TDRE) bit is set to "1" when the transmission interrupt enable bit (TIE) contains "1". At this time, the next piece of transmit data can be written to the UART/SIO serial output data register (TDR0). Serial transmission can be continued with the transmission operation enable bit (TXE) set to "1".

To use a transmission completion interrupt to detect the completion of serial transmission, enable transmission completion interrupt output this way: TEIE = 0, TCIE = 1. Upon completion of transmission, the transmission completion flag (TCPL) is set to "1" and a transmission completion interrupt occurs.

**Figure 20.7-15 8-bit Transmission in Synchronous CLK Mode**



- Concurrent transmission and reception

- <When external clock is enabled>**

- Transmission and reception can be performed independently of each other. Transmission and reception can therefore be performed at the same time or even when their phases are shifted from each other and overlapping.

- <When internal clock is enabled>**

- As the transmitting side generates a serial clock, reception is influenced.

- If transmission stops during reception, the receiving side is suspended. It resumes reception when the transmitting side is restarted.

- Refer to "20.4 Pins of UART/SIO" for operation with serial clock output and operation with serial clock input.



## 20.8 Sample Programs for UART/SIO

We provide sample programs that can be used to operate UART/SIO.

### ■ Sample Programs for UART/SIO

For information about the sample programs for UART/SIO, refer to "■ Sample Programs" in Preface.

### ■ Setting Methods not Covered by Sample Programs

#### ● How to select the operation mode

The operation mode select bit (SMC10.MD) is used.

Operation mode		Operation mode selection (MD)
Mode 0	Asynchronous clock mode (UART)	Set the bit to "0"
Mode 1	Synchronous clock mode (SIO)	Set the bit to "1"

#### ● Operation clock types and how to select it

The clock select bit (SMC10.CKS) is used.

Clock input	Clock selection (CKS)
To select a dedicated baud rate generator	Set the bit to "0"
To select an external clock	Set the bit to "1"

#### ● How to use UCK0, UI0, and UO0 pin

Uses the following setting.

	UART
To set the UCK0 pin as input	DDR1.P12 = 0 SMC20:SCKE = 0
To set the UCK0 pin as output	SMC20:SCKE = 1
When using UI0 pin	DDR1.P10 = 0
When using UO0 pin	SMC20:TXOE = 1

● How to enable/stop UART operation

The reception operation enable bit (SMC20.RXE) is used.

Control item	Reception interrupt enable bit (RXE)
Disabling (stopping) reception	Set the bit to "0".
Enabling reception	Set the bit to "1".

The transmission operation control bit (SMC20.TXE) is used.

Control item	Transmission operation enable bit (TXE)
Disabling (stopping) transmission	Set the bit to "0".
Enabling transmission	Set the bit to "1".

● How to set the parity

The parity control (SMC10.PEN) and parity polarity (SMC10.TDP) bits are used.

Operation	Parity control (PEN)	Parity polarity (TDP)
To set to no parity	Set the bit to "0".	–
To set to even parity	Set the bit to "1".	Set the bit to "0".
To set to odd parity	Set the bit to "1".	Set the bit to "1".

● How to set the data length

The data length select bit (SMC10.CBL[1:0]) is used.

Operation	Data length select bit (CBL[1:0])
To set the bit length to 5	Set the bits to "00 <sub>B</sub> ".
To set the bit length to 6	Set the bits to "01 <sub>B</sub> ".
To set the bit length to 7	Set the bits to "10 <sub>B</sub> ".
To set the bit length to 8	Set the bits to "11 <sub>B</sub> ".

● How to select the STOP bit length

The STOP bit length control bit (SMC10.SBL) is used.

Operation	STOP bit length control (SBL)
To set STOP bit length to 1	Set the bit to "0".
To set STOP bit length to 2	Set the bit to "1".

● How to clear the error flag

The reception error flag clear bit (SMC20.RERC) is used.

Control item	Reception error flag clear bit (RERC)
When clearing error flags (PER, OVE, FER)	Set the bit to "0".

● How to set the transfer direction

The serial data direction control bit (SMC10.BDS) is used.

LSB first/MSB first can be selected for transfer direction in any operation mode.

Control item	Serial data direction control (BDS)
When selecting LSB first transfer (from least significant bit)	Set the bit to "0".
When selecting MSB first transfer (from most significant bit)	Set the bit to "1".

● How to clear the reception completion flag

Uses the following setting.

Control item	Method
To clear the reception completion flag	Read the RDR0 register

The first RDR0 register read is the reception initiation.

● How to clear the transmit buffer empty flag

Uses the following setting.

Control item	Method
To clear the transmit buffer empty flag	Write to TDR0 register

The first TDR0 register write is the transmit initiation.

● How to set the baud rate

See Section "20.7.1 Operating Description of Operation Mode 0".

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

Channel	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR1) Address: 0007A <sub>H</sub>	#4 Address: 0FFF2 <sub>H</sub>

● Enabling, disabling, and clearing interrupts

The interrupt request enable bits (SMC20:RIE), (SMC20:TCIE), (SMC20:TEIE) are used to enable interrupts.

	UART reception	UART transmission	
	Reception interrupt enable bit (RIE)	Transmission completion interrupt enable bit (TCIE)	Transmission data register empty interrupt enable bit (TEIE)
To disable interrupt requests	Set to "0"		
To enable interrupt requests	Set to "1"		

The following setting is used to clear interrupt requests.

	UART reception	UART transmission
To clear interrupt requests	Read from serial input register (RDR 0) to clear reception data register full bit (RDRF).	The transmit data register empty (TDRE) is set to "0" by writing data to the serial output data register (TDR0).
	Write "0" to error flag clear bit (RERC) to clear error flags (PER, OVE, FER) to "0".	



# ***CHAPTER 21***

---

# ***UART/SIO DEDICATED BAUD RATE GENERATOR***

**This chapter describes the functions and operations of the dedicated baud rate generator of UART/SIO.**

- 21.1 Overview of UART/SIO Dedicated Baud Rate Generator
- 21.2 Channels of UART/SIO Dedicated Baud Rate Generator
- 21.3 Registers of UART/SIO Dedicated Baud Rate Generator
- 21.4 Operating Description of UART/SIO Dedicated Baud Rate Generator

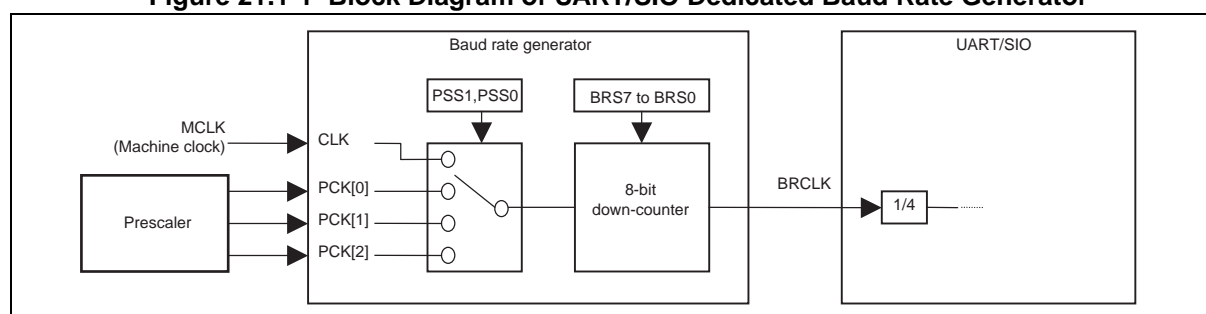
## 21.1 Overview of UART/SIO Dedicated Baud Rate Generator

The UART/SIO dedicated baud rate generator generates the baud rate for the UART/SIO.

The generator consists of the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

### ■ Block Diagram of UART/SIO Dedicated Baud Rate Generator

Figure 21.1-1 Block Diagram of UART/SIO Dedicated Baud Rate Generator



### ■ Input Clock

The UART/SIO dedicated baud rate generator uses the output clock from the prescaler or the machine clock as its input clock.

### ■ Output Clock

The UART/SIO dedicated baud rate generator supplies its clock to the UART/SIO.

## 21.2 Channels of UART/SIO Dedicated Baud Rate Generator

---

This section describes the channels of the UART/SIO dedicated baud rate generator.

---

### ■ Channels of UART/SIO Dedicated Baud Rate Generator

This series contains one channel of the UART/SIO dedicated baud rate generator.

Table 21.2-1 shows the registers of the UART/SIO dedicated baud rate generator.

**Table 21.2-1 Registers of UART/SIO Dedicated Baud Rate Generator**

Channel	Register name	Corresponding register (Representation in this manual)
0	PSSR0	UART/SIO dedicated baud rate generator prescaler selection register
	BRSR0	UART/SIO dedicated baud rate generator baud rate setting register



## 21.3 Registers of UART/SIO Dedicated Baud Rate Generator

The registers related to the UART/SIO dedicated baud rate generator are namely the UART/SIO dedicated baud rate generator prescaler selection register (PSSR0) and UART/SIO dedicated baud rate generator baud rate setting register (BRSR0).

### ■ Registers Related to UART/SIO Dedicated Baud Rate Generator

Figure 21.3-1 Registers Related to UART/SIO Dedicated Baud Rate Generator

UART/SIO dedicated baud rate generator prescaler selection register (PSSR0)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PSSR0 0FBE <sub>H</sub>	—	—	—	—	—	BRGE	PSS1	PSS0	00000000 <sub>B</sub>
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	R/W	
UART/SIO dedicated baud rate generator baud rate setting register (BRSR0)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
BRSR0 0FBF <sub>H</sub>	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable (Read value is the same as write value)									
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)									

## 21.3.1 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

The UART/SIO dedicated baud rate generator prescaler register (PSSR0) controls the output of the baud rate clock and the prescaler.

### ■ UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

Figure 21.3-2 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

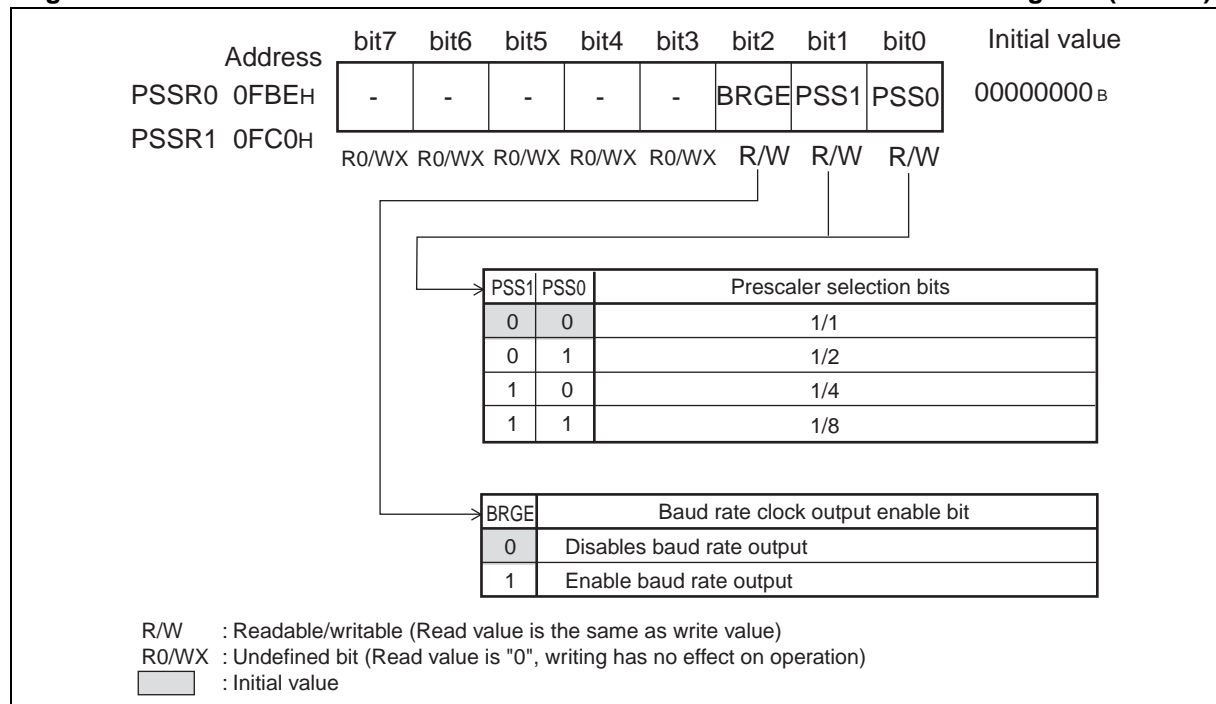


Table 21.3-1 UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0)

Bit name		Function															
bit7 to bit3	Undefined bits	These bits are undefined. Reading the bits always returns "0".															
bit2	BRGE: Baud rate clock output enable bit	This bit enables the output of the baud rate clock "BRCLK". <b>When set to "1"</b> : loads BRS[7:0] to the 8-bit down-counter and outputs "BRCLK", which is supplied to the UART/SIO. <b>When set to "0"</b> : stops the output of "BRCLK".															
bit1, bit0	PSS1, PSS0: Prescaler selection bits	<table border="1"> <thead> <tr> <th>PSS1</th><th>PSS0</th><th>Prescaler selection</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1/1</td></tr> <tr> <td>0</td><td>1</td><td>1/2</td></tr> <tr> <td>1</td><td>0</td><td>1/4</td></tr> <tr> <td>1</td><td>1</td><td>1/8</td></tr> </tbody> </table>	PSS1	PSS0	Prescaler selection	0	0	1/1	0	1	1/2	1	0	1/4	1	1	1/8
PSS1	PSS0	Prescaler selection															
0	0	1/1															
0	1	1/2															
1	0	1/4															
1	1	1/8															

## 21.3.2 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

The UART/SIO dedicated baud rate generator baud rate setting register (BRSR0) controls the baud rate settings.

### ■ UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

Figure 21.3-3 UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
BRSR0 0FBF <sub>H</sub>	BRS7	BRS6	BRS5	BRS4	BRS3	BRS2	BRS1	BRS0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable/writable (Read value is the same as write value)

This register sets the cycle of the 8-bit down-counter. This register can be used to set any baud rate clock. Write to the register when the UART is stopped.

Do not set BRS[7:0] to "00<sub>H</sub>" or "01<sub>H</sub>" in clock asynchronous mode.

## 21.4 Operating Description of UART/SIO Dedicated Baud Rate Generator

---

The UART/SIO dedicated baud rate generator serves as the baud rate generator for asynchronous clock mode.

---

### ■ Baud Rate Setting

The SMC10 register (CKS bit) of the UART/SIO is used to select the serial clock. This selects the UART/SIO dedicated baud rate generator.

In asynchronous CLK mode, the shift clock that is selected by the CKS bit and divided by four is used and transfers can be performed within the range from -2% to +2%. The baud rate calculation formula for the UART/SIO dedicated baud rate generator is shown below.

**Figure 21.4-1 Baud Rate Calculation Formula when UART/SIO Dedicated Baud Rate Generator Is Used**

$\text{Baud rate} = \frac{\text{Machine clock (MCLK)}}{4 \times \begin{matrix} \boxed{1} \\ \boxed{2} \\ \boxed{4} \\ \boxed{8} \end{matrix} \times \begin{matrix} \boxed{2} \\ \boxed{:} \\ \boxed{255} \end{matrix}} \text{ [bps]}$
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">             UART prescaler selection register (PSSR0)              Prescaler selection              (PSS1, PSS0)           </div> <div style="text-align: center;">             UART baud rate setting register (BRSR0)              Baud Rate Setting              (BRS7 to BRS0)           </div> </div>

**Table 21.4-1 Sample Asynchronous Transfer Rates by Baud Rate Generator  
(Machine Clock = 10MHz, 16MHz, 16.25MHz)**

Settings of UART/SIO dedicated baud rate generator		UART Internal division	Total division ratio (PSS × BRS × 4)	Baud rate (10MHz/ Total division ratio)	Baud rate (16MHz/ Total division ratio)	Baud rate (16.25MHz/ Total division ratio)
Prescaler selection PSS[1:0]	Baud rate counter setting BRS[7:0]					
1 (Setting value:0, 0)	20	4	80	125000	200000	203125
1 (Setting value:0, 0)	22	4	88	113636	181818	184659
1 (Setting value:0, 0)	44	4	176	56818	90909	92330
1 (Setting value:0, 0)	87	4	348	28736	45977	46695
1 (Setting value:0, 0)	130	4	520	19231	30769	31250
2 (Setting value:0, 1)	130	4	1040	9615	15385	15625
4 (Setting value:1, 0)	130	4	2080	4808	7692	7813
8 (Setting value:1, 1)	130	4	4160	2404	3846	3906

The baud rate can be set in UART mode within the following range.

**Table 21.4-2 Permissible Baud Rate Range in UART Mode**

PSS[1:0]	BRS[7:0]
00 <sub>B</sub> to 11 <sub>B</sub>	02 <sub>H</sub> (2) to FF <sub>H</sub> (255)

# **CHAPTER 22**

---

## ***LIN-UART***

**This chapter describes the function and operation of the LIN-UART**

- 22.1 Overview of LIN-UART
- 22.2 Configuration of LIN-UART
- 22.3 Pins of LIN-UART
- 22.4 Registers of LIN-UART
- 22.5 Interrupt of LIN-UART
- 22.6 LIN-UART Baud Rate
- 22.7 Operations and Setup Procedure Example of LIN-UART
- 22.8 Notes on Using LIN-UART
- 22.9 Sample Programs of LIN-UART

## 22.1 Overview of LIN-UART

The LIN (Local Interconnect Network)-UART is a general-purpose serial data communication interface for synchronous or asynchronous (start-stop synchronization) communication with external devices. In addition to a bi-directional communication function (normal mode) and master/slave communication function (multiprocessor mode: supports both master and slave operation), the LIN-UART also supports the special functions used by the LIN bus.

### ■ Functions of LIN-UART

The LIN-UART is a general-purpose serial data communication interface for transmitting serial data to and receiving data from other CPUs and peripheral devices. Table 22.1-1 lists the functions of the LIN-UART.

**Table 22.1-1 Functions of LIN-UART**

	Function
Data buffer	Full-duplex double buffer
Serial input	The LIN-UART oversamples received data for five times to determine the received value by majority (only asynchronous mode).
Transfer mode	<ul style="list-style-type: none"> <li>• Clock synchronization (Select start/stop synchronization, or start/stop bit)</li> <li>• Clock asynchronous (Start/stop bits available)</li> </ul>
Baud rate	<ul style="list-style-type: none"> <li>• Dedicated baud rate generator provided (made of a 15-bit reload counter)</li> <li>• The external clock can be inputted. The reload counter can also be used to adjust the external clock.</li> </ul>
Data length	<ul style="list-style-type: none"> <li>• 7 bits (not in synchronous or LIN mode)</li> <li>• 8 bits</li> </ul>
Signaling	NRZ (Non Return to Zero)
Start bit timing	Synchronization with the start bit falling edge in asynchronous mode.
Reception error detection	<ul style="list-style-type: none"> <li>• Framing error</li> <li>• Overrun error</li> <li>• Parity error (Not supported in operation mode 1)</li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• Reception interrupts (reception completed, reception error detected, LIN synch break detected)</li> <li>• Transmit interrupts (send data empty)</li> <li>• Interrupt requests to THIO (LIN synch field detected: LSYN)</li> </ul>
Master/slave mode communication function (Multiprocessor mode)	Capable of 1 (master) to n (slaves) communication (support both the master and slave system)
Synchronous Mode	Send side/receive side of serial clock
Pin access	Serial I/O pin states can be read directly.
LIN bus option	<ul style="list-style-type: none"> <li>• Master device operation</li> <li>• Slave device operation</li> <li>• LIN synch break detection</li> <li>• LIN Synch break generation</li> <li>• Detection of LIN synch field start/stop edges connected to the 8/16-bit compound timer</li> </ul>
Synchronous serial clock	Continuous output to the SCK pin is possible for synchronous communication using the start/stop bits
Clock delay option	Special synchronous clock mode for delaying the clock (used for serial peripheral interface (SPI))

The LIN-UART has four operation modes. The operation mode is selected by the MD0 and MD1 bits in the LIN-UART serial mode register (SMR). Mode 0 and mode 2 are used for bi-directional serial communication; mode 1 for master/slave communication; and mode 3 for LIN master/slave communication.

**Table 22.1-2 LIN-UART Operation Modes**

Operation mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multi processor mode	7 bits or 8 bits +1*	–	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	
3	LIN mode	8 bits	–	Asynchronous	1 bit	LSB first

–: Unavailable

\*: "+1" is the address/data selection bit (AD) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operation modes.

**Table 22.1-3 LIN-UART Operation Modes**

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

- Mode 1 supports both master and slave operation for the multiprocessor mode.
- Mode 3 is fixed to communication format 8-bit data, no parity, 1 stop bit, LSB-first.



## **22.2 Configuration of LIN-UART**

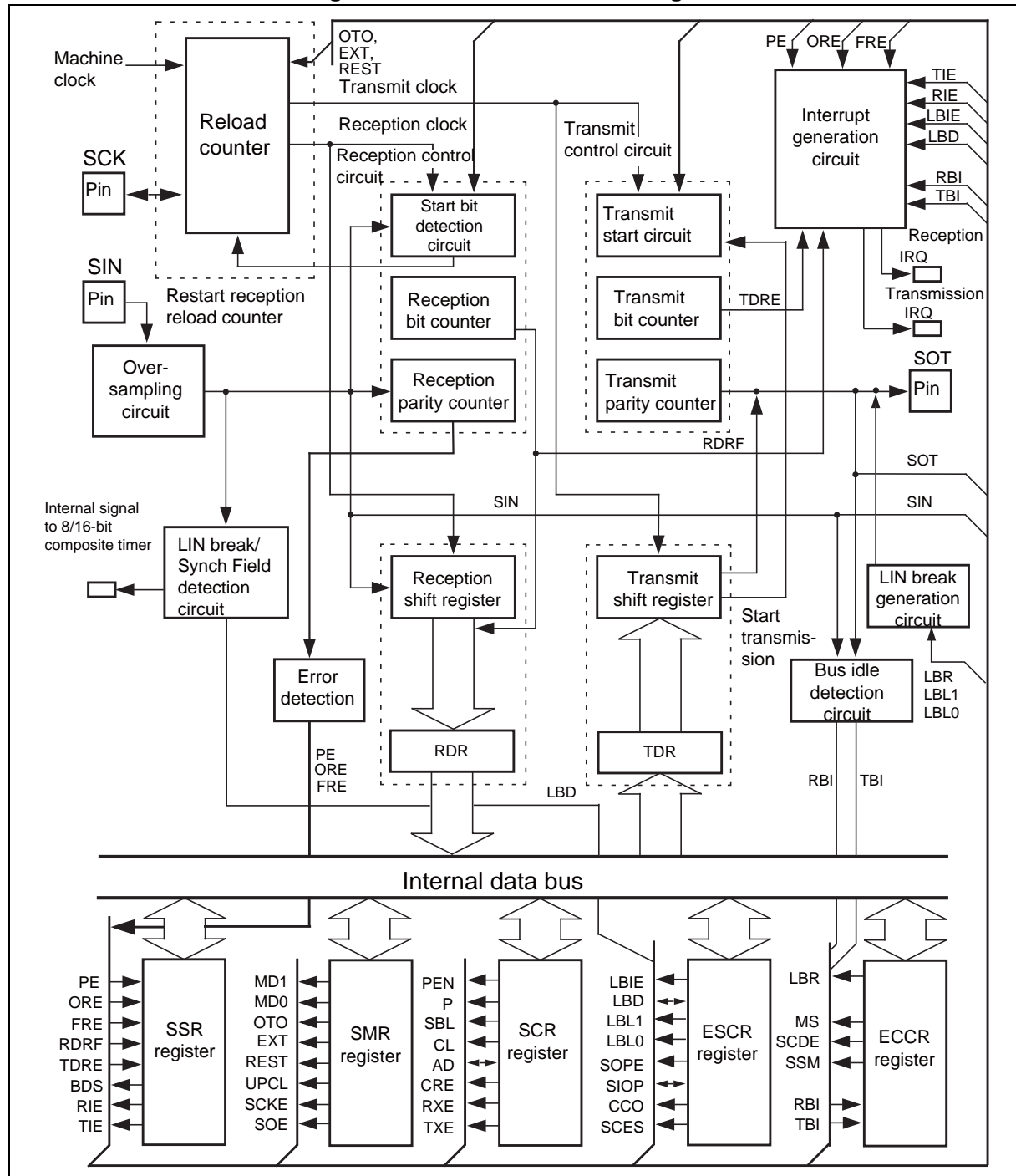
---

**LIN-UART is made up of the following blocks.**

- **Reload Counter**
  - **Reception control circuit**
  - **Reception shift register**
  - **LIN-UART reception data register (RDR)**
  - **Transmit control circuit**
  - **Transmit shift register**
  - **LIN-UART transmit data register (TDR)**
  - **Error detection circuit**
  - **Oversampling circuit**
  - **Interrupt generation circuit**
  - **LIN synch break/Synch Field detection circuit**
  - **Bus idle detection circuit**
  - **LIN-UART serial control register (SCR)**
  - **LIN-UART serial mode register (SMR)**
  - **LIN-UART serial status register (SSR)**
  - **LIN-UART extended status control register (ESCR)**
  - **LIN-UART extended communication control register (ECCR)**
-

■ LIN-UART Block Diagram

Figure 22.2-1 LIN-UART Block Diagram



● Reload counter

This block is a 15-bit reload counter serving as a dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/reception clock from the external or internal clock. The count value in the transmit reload counter is read from the baud rate generator 1, 0 (BGR1, BGR0).

● Reception control circuit

This block consists of a reception bit counter, a start bit detection circuit, and a reception parity counter. The reception bit counter counts the reception data bits and sets a flag in the LIN-UART reception data register when one data reception is completed according to the specified data length. If the reception interrupt is enabled at this time, a reception interrupt request is generated. The start bit detection circuit detects a start bit in a serial input signal. When a start bit is detected, the circuit sends a signal to the reload counter in synchronization with the start bit falling edge. The reception parity counter calculates the parity of the received data.

● Reception shift register

The circuit inputs received data from the SIN pin while bit-shifting and transfers it to the RDR register upon completion of reception.

● LIN-UART Reception Data Register (RDR)

This register retains the received data. Serial input data is converted and stored in the LIN-UART reception data register.

● Transmit control circuit

This block consists of a transmit bit counter, a transmission start circuit, and a transmit parity counter. The transmit bit counter counts the transmit data bits and sets a flag in the transmit data register when one data transmission is completed according to the specified data length. If the transmit interrupt is enabled at this time, a transmit interrupt request is generated. The transmit start circuit starts transmission when data is written to the TDR. The transmit parity counter generates a parity bit for data to be transmitted if the data is parity-checked.

● Transmit shift register

The data written to the LIN-UART transmit data register (TDR) is transferred to the transmit shift register, and output to the SOT pin during bit-shifting.

● LIN-UART transmit data register (TDR)

This register sets the transmit data. The written data is converted to serial data and output.

● Error detection circuit

This circuit detects an error upon completion of reception, if any. If an error occurs, the corresponding error flag is set.

● Oversampling circuit

In asynchronous mode, the LIN-UART oversamples received data for five times to determine the received value by majority. The LIN-UART stops during operation in synchronous mode.

● Interrupt generation circuit

This circuit controls all interrupt factors. An interrupt is generated immediately if the corresponding interrupt enable bit has been set.

● LIN synch break/Synch Field detection circuit

This circuit detects a LIN synch break when the LIN master node transmits a message header. The LBD flag is set when the LIN synch break is detected. An internal signal is output to 8/16-bit compound timer in order to detect the first and fifth falling edges of the LIN synch Field and to measure the actual serial clock synchronization transmitted by the master node.

**● LIN synch break generation circuit**

This circuit generates a LIN synch break with the specified length.

**● Bus idle detection circuit**

This circuit detects that no transmission or reception is in progress, and generates the TBI and RBI flag bits.

**● LIN-UART serial control register (SCR)**

Operating functions are as follows:

- Sets parity bit existence
- Parity bit selection
- Sets stop bit length
- Sets data length
- Selects the frame data format in mode 1
- Clears error flag
- Enables/disables transmission
- Enables/disables reception

**● LIN-UART serial mode register (SMR)**

Operating functions are as follows:

- Selects the LIN-UART operation mode
- Selects a clock input source
- Selects between one-to-one connection or reload counter connection for the external clock
- Resets a dedicated reload timer
- LIN-UART software reset (maintains register settings)
- Enables/disables output to the serial data pin
- Enables/disables output to the clock pin

**● LIN-UART serial status register (SSR)**

Operating functions are as follows:

- Check transmission/reception or error status
- Selects the transfer direction (LSB-first or MSB-first)
- Enables/disables reception interrupts
- Enables/disables transmit interrupts

**● Extended status control register (ESCR)**

- Enables/disables LIN synch break interrupts
- LIN synch break detection
- Selects LIN synch break length
- Direct access to SIN pin and SOT pin
- Sets continuous clock output in LIN-UART synchronous clock mode
- Sampling clock edge selection

● LIN-UART extended communication control register (ECCR)

- Bus idle detection
- Synchronous clock setting
- LIN synch break generation

■ **Input Clock**

LIN-UART uses a machine clock or an input signal from the SCK pin as an input clock.

Input clock is used as clock source of transmission/reception of LIN-UART.

## MB95110B/M Series

### 22.3 Pins of LIN-UART

This section describes LIN-UART pins.

#### ■ Pins related to LIN-UART

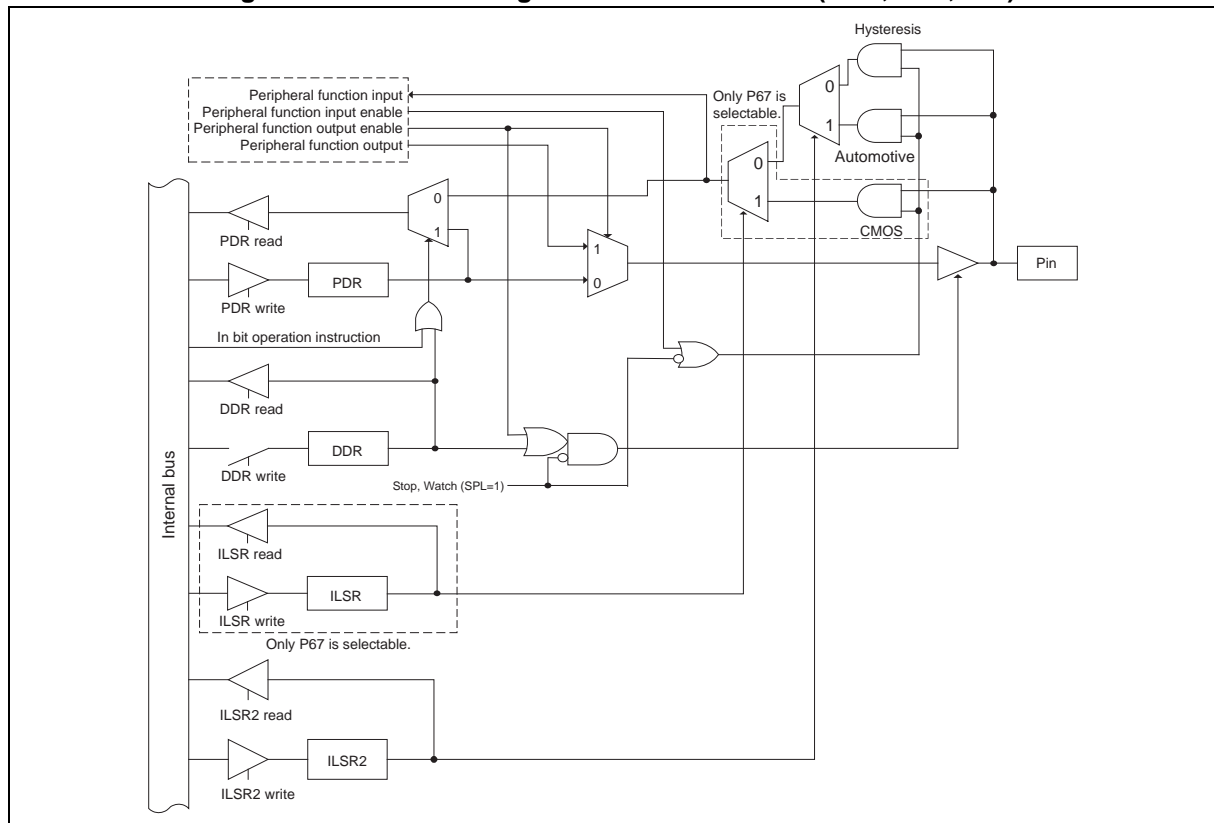
The LIN-UART pins also serve as general-purpose ports. Table 22.3-1 lists the LIN-UART pin.

**Table 22.3-1 LIN-UART Pins**

Pin name	Pin function	Required settings for using the pin
SIN	Serial data input	Set to the input port (DDR: corresponding bit = 0)
SOT	Serial data output	Set to output enable (SMR:SOE = 1)
SCK	Serial clock input/output	Set to the input port when used as clock input (DDR: corresponding bit = 0)
		Set to output enable when used as clock output (SMR:SCKE = 1)

#### ■ Block Diagram of LIN-UART Pins

**Figure 22.3-1 Block Diagram of LIN-UART Pins (SCK, SOT, SIN)**



## 22.4 Registers of LIN-UART

This section lists the registers of LIN-UART.

### ■ Register List of LIN-UART

Figure 22.4-1 Register List of LIN-UART

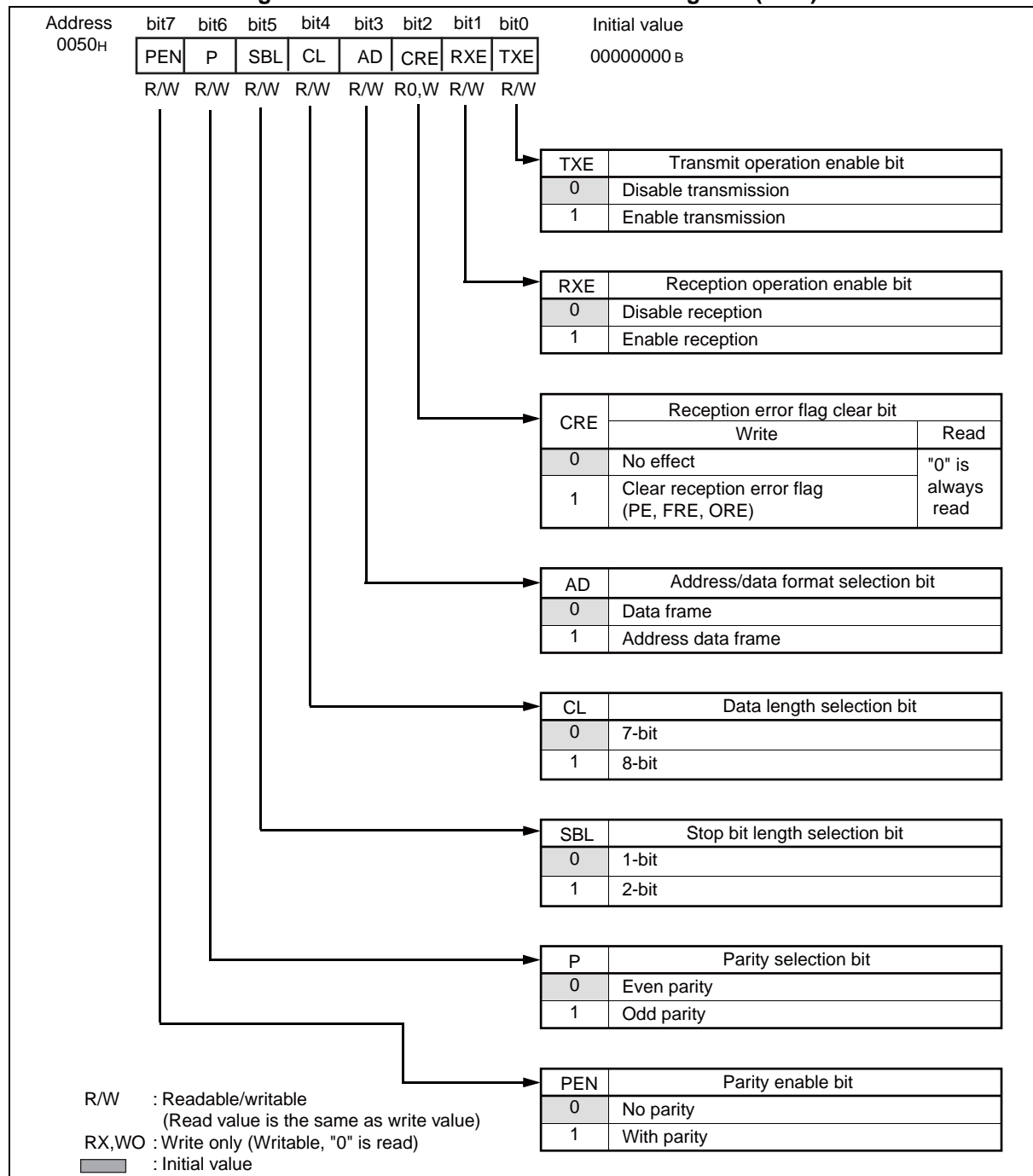
LIN-UART serial control register (SCR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0050 <sub>H</sub>	PEN	P	SBL	CL	AD	CRE	RXE	TXE
	R/W	R/W	R/W	R/W	R/W	R0,W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
LIN-UART serial mode register (SMR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0051 <sub>H</sub>	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
	R/W	R/W	R/W	R/W	R0,W	R0,W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
LIN-UART serial status register (SSR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0052 <sub>H</sub>	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE
	R/WX	R/WX	R/WX	R/WX	R/WX	R/W	R/W	R/W
Initial value								
00001000 <sub>B</sub>								
LIN-UART reception data register/LIN-UART transmit data register (RDR/TDR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0053 <sub>H</sub>	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
LIN-UART extended status control register (ESCR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0054 <sub>H</sub>	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES
	R/W	R(RM1),W	R/W	R/W	R/W	R(RM1),W	R/W	R/W
Initial value								
00000100 <sub>B</sub>								
LIN-UART extended communication control register (ECCR)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0055 <sub>H</sub>	Reserv ed	LBR	MS	SCDE	SSM	Reserv ed	RBI	TBI
	RX,W0	R0,W	R/W	R/W	R/W	RX,W0	R/WX	R/WX
Initial value								
000000XX <sub>B</sub>								
LIN-UART baud rate generator register 1 (BGR1)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FBC <sub>H</sub>	—	BGR14	BGR13	BGR12	BGR11	BGR10	BGR9	BGR8
	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
LIN-UART baud rate generator register 0 (BGR0)								
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0FBD <sub>H</sub>	BGR7	BGR6	BGR5	BGR4	BGR3	BGR2	BGR1	BGR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value								
00000000 <sub>B</sub>								
R/W : Readable/writable (Read value is the same as write value)								
R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)								
R/WX : Read only (Readable, writing has no effect on operation)								
R0, W : Write only (Writable, "0" is read)								
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)								
RX,W0 : Reserved bit (Write value is indeterminate, write value is "0")								

## 22.4.1 LIN-UART Serial Control Register (SCR)

The LIN-UART serial control register (SCR) is used to set parity, select the stop bit length and data length, select the frame data format in mode 1, clear the reception error flag, and enable/disable transmission/reception.

### ■ LIN-UART Serial Control Register (SCR)

Figure 22.4-2 LIN-UART Serial Control Register (SCR)





**Table 22.4-1 Functions of Each Bit in LIN-UART Serial Control Register (SCR)**

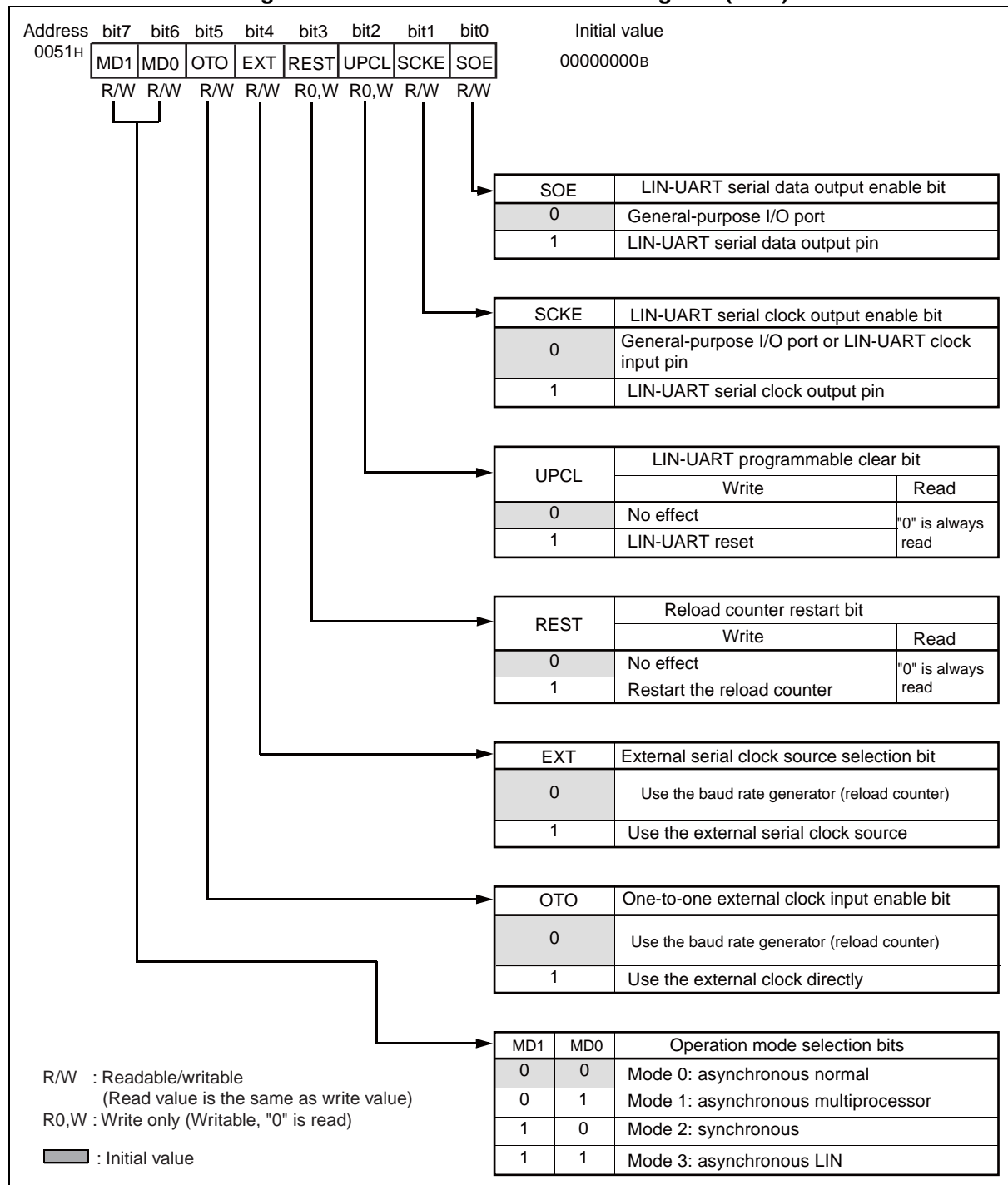
Bit name		Function
bit7	PEN: Parity enable bit	Specify whether or not to add (at transmission) and detect (at reception) a parity bit. Note: The parity bit is added only in operation mode 0, or in operation mode 2 with the settings that start/stop is set (ECCR:SSM=1). This bit is fixed to "0" in mode 3 (LIN).
bit6	P: Parity selection bit	Set either odd parity (1) or even parity (0) if the parity bit has been selected (SCR:PEN = 1).
bit5	SBL: Stop bit length selection bit	Set the bit length of the stop bit (frame end mark in transmit data) in operation mode 0, 1 (asynchronous) or in operation mode 2 (synchronous) with the settings that start/stop bit is set (ECCR:SSM=1). This bit is fixed to "0" in mode 3 (LIN).
bit4	CL: Data length selection bit	Specify the data length to be transmitted and received. This bit is fixed to "1" in mode 2 and mode 3.
bit3	AD: Address/Data format selection bit	Specify the data format for the frame to be transmitted and received in multiprocessor mode (mode 1). Write to this bit in master mode; read this bit in slave. The operation in master mode is as follows. "0": Set to data frame. "1": Set to address data frame. The value of last received data format is read. Note: See Section "22.8 Notes on Using LIN-UART" for using this bit.
bit2	CRE: Reception error flag clear bit	This bit is to clear FRE, ORE, and PE flags in serial status register (SSR). "0": No effect. "1": Clear the error flag. Reading this bit always returns "0". Note: Disable the reception operation (RXE=0) first, and then clear the reception error flags. If an error flag is cleared before the reception operation is disabled, the reception will be aborted at that time, and resume later. This may result in a data reception error.
bit1	RXE: Reception operation enable bit	Enable or disable the reception of LIN-UART. "0": Disable data frame reception. "1": Enable data frame reception. The LIN synch break detection in mode 3 is not affected. Note: When the reception is disabled (RXE = 0) during reception, the reception halts immediately. In that case, the data is not guaranteed.
bit0	TXE: Transmit operation enable bit	Enable or disable the transmission of LIN-UART. "0": Disable data frame transmission. "1": Enable data frame transmission. Note: When the transmission is disabled (TXE = 0) during transmission, the transmission halts immediately. In that case, the data is not guaranteed.

## 22.4.2 LIN-UART Serial Mode Register (SMR)

The LIN-UART serial mode register (SMR) is used to select the operation mode, specify the baud rate clock, and enable/disable output to the serial data and clock pins.

### ■ LIN-UART Serial Mode Register (SMR)

Figure 22.4-3 LIN-UART Serial Mode Register (SMR)



**Table 22.4-2 Functions of Each Bit in LIN-UART Serial Mode Register (SMR)**

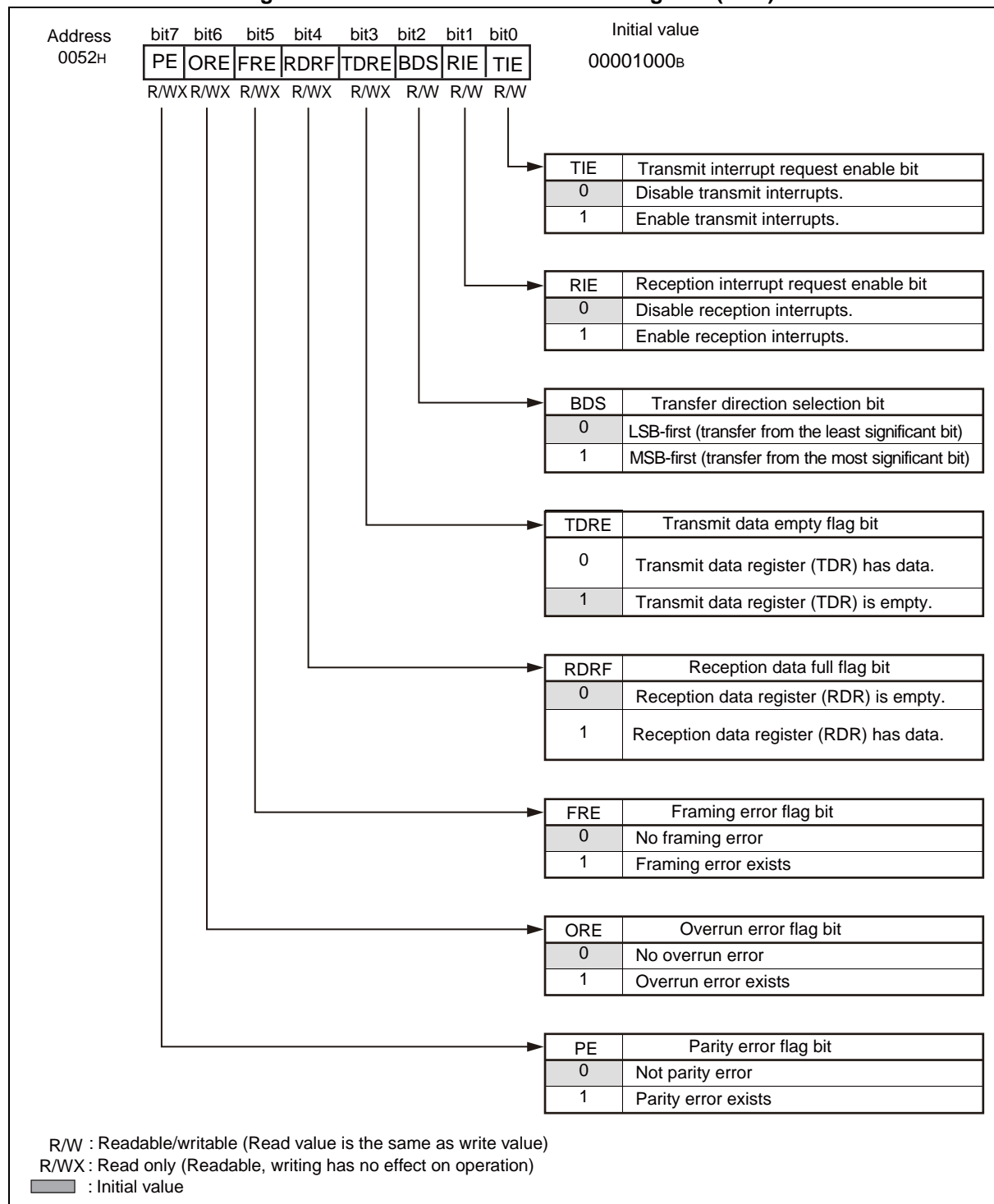
Bit name		Function																				
bit7, bit6	MD1, MD0: Operation mode select bits	Set the operation mode. Note: If the mode is changed during communication, the transmission/ reception of LIN-UART halts and waits for starting the next communication.																				
		<table><tr><td>MD1</td><td>MD0</td><td>Mode</td><td>Type</td></tr><tr><td>0</td><td>0</td><td>0</td><td>Asynchronous (Normal mode)</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Asynchronous (Multiprocessor mode)</td></tr><tr><td>1</td><td>0</td><td>2</td><td>Synchronous (Normal mode)</td></tr><tr><td>1</td><td>1</td><td>3</td><td>Asynchronous (LIN mode)</td></tr></table>	MD1	MD0	Mode	Type	0	0	0	Asynchronous (Normal mode)	0	1	1	Asynchronous (Multiprocessor mode)	1	0	2	Synchronous (Normal mode)	1	1	3	Asynchronous (LIN mode)
		MD1	MD0	Mode	Type																	
		0	0	0	Asynchronous (Normal mode)																	
		0	1	1	Asynchronous (Multiprocessor mode)																	
		1	0	2	Synchronous (Normal mode)																	
1	1	3	Asynchronous (LIN mode)																			
bit5	OTO: One-to-one external clock input enable bit	"1": Enable the external clock to be used directly as the LIN-UART serial clock. Used for reception side of serial clock (ECCR:MS = 1) in operation mode 2 (synchronous). When EXT = 0, the OTO bit is fixed to "0".																				
bit4	EXT: External serial clock source select bit	Select a clock input. "0": Select the clock of the internal baud rate generator (reload counter). "1": Select the external serial clock source.																				
bit3	REST: Reload counter restart bit	Restart the reload counter. "0": No effect. "1": Restart the reload counter. Reading this bit always returns "0".																				
bit2	UPCL: LIN-UART programmable clear bit (Reset the LIN- UART software reset)	Reset the LIN-UART. "0": No effect. "1": Reset the LIN-UART immediately (LIN-UART software reset). However, the register settings are maintained. At that time, transmission and reception are halted. All of the transmit/reception interrupt factors (TDRE, RDRF, LBD, PE, ORE, FRE) are reset. Reset the LIN-UART after the interrupt and transmission are disabled. Also, the reception data register is cleared (RDR = 00 <sub>H</sub> ), and the reload counter is restarted. Reading this bit always returns "0".																				
bit1	SCKE: LIN-UART serial clock output enable bit	Control the serial clock I/O port. "0": The SCK pin works as a general-purpose I/O port or a serial clock input pin. "1": This pin works as the serial clock output pin and outputs the clock in operation mode 2 (synchronous). Note: When the SCK pin is used as a serial clock input (SCKE = 0), set the corresponding DDR bits in the general-purpose I/O port as an input port. Also, select the external clock (EXT = 1) by using the clock select bit. When the SCK pin is set as a serial clock output (SCKE = 1), this pin works as a serial clock output pin regardless of the state of the general-purpose I/O port.																				
bit0	SOE: LIN-UART serial data output enable bit	Enable or disable output of serial data. "0": The SOT pin works as a general-purpose I/O port. "1": The SOT pin works as a serial data output pin (SOT). When set as a serial data output (SOE = 1), the SOT pin works as a SOT pin regardless of a general-purpose I/O port.																				

### 22.4.3 LIN-UART Serial Status Register (SSR)

The LIN-UART serial status register (SSR) is used to check the status of transmission/reception or error, and to enable/disable interrupts.

#### ■ LIN-UART Serial Status Register (SSR)

Figure 22.4-4 LIN-UART serial status register (SSR)



**Table 22.4-3 Functions of Each Bit in serial status register (SSR)**

Bit name		Function
bit7	PE: Parity error flag bit	<p>Detect a parity error in received data.</p> <ul style="list-style-type: none"> <li>• This bit is set to "1" when a parity error occurs during reception with PEN = 1, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR).</li> <li>• Output a reception interrupt request when both PE bit and RIE bit are "1".</li> <li>• When this flag is set, the data in the reception data register (RDR) is invalid.</li> </ul>
bit6	ORE: Overrun error flag bit	<p>Detect an overrun error in received data.</p> <ul style="list-style-type: none"> <li>• This bit is set to "1" when an overrun occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR).</li> <li>• Output a reception interrupt request when both ORE bit and RIE bit are "1".</li> <li>• When this flag is set, the data in the reception data register (RDR) is invalid.</li> </ul>
bit5	FRE: Framing error flag bit	<p>Detect a framing error in received data.</p> <ul style="list-style-type: none"> <li>• This bit is set to "1" when a framing error occurs during reception, and cleared by writing "1" to the CRE bit in the LIN-UART serial control register (SCR).</li> <li>• Output a reception interrupt request when both FRE bit and RIE bit are "1".</li> <li>• When this flag is set, the data in the reception data register (RDR) is invalid.</li> </ul>
bit4	RDRF: Reception data full flag bit	<p>This flag shows the status of the reception data register (RDR).</p> <ul style="list-style-type: none"> <li>• This bit is set to "1" when received data is loaded into the reception data register (RDR), and cleared to "0" by reading RDR.</li> <li>• Output a reception interrupt request when both RDRF bit and RIE bit are "1".</li> </ul>
bit3	TDRE: Transmit data empty flag bit	<p>This flag shows the status of the transmit data register (TDR).</p> <ul style="list-style-type: none"> <li>• This bit is set to "0" by writing the transmit data to TDR, and indicates that the TDR has valid data. This bit is set to "1" when data is loaded into the transmit shift register and the transmission starts, and indicates that the TDR does not have effective data.</li> <li>• Output a transmit interrupt request when both TDRE bit and TIE bit are "1".</li> <li>• When the TDRE bit is "1", setting the LBR bit in the extended communication control register (ECCR) to "1" changes the TDRE bit to "0". Then, the TDRE bit goes back to "1" after LIN sync break is generated.</li> </ul> <p>Note: The initial state is TDRE = 1.</p>
bit2	BDS: Transfer direction selection bit	<p>Specify whether the transfer serial data is transfer from the least significant bit (LSB-first, BDS = 0) or from the most significant bit (MSB-first, BDS = 1).</p> <p>Note: Since data values are exchanged between the upper and lower when the data is read/written to the serial data register, changing BDS bit after writing data to the RDR register invalidates the written data. The BDS bit is fixed to "0" in mode 3 (LIN).</p>
bit1	RIE: Reception interrupt request enable bit	<p>Enable or disable the reception interrupt request output to the interrupt controller.</p> <p>Output a reception interrupt request when both the RIE bit and the reception data flag bit (RDRF) are "1", or when one or more error flag bits (PE, ORE, FRE) is "1".</p>
bit0	TIE: Transmit interrupt request enable bit	<p>Enable or disable the transmit interrupt request output to the interrupt controller.</p> <p>Output a transmit interrupt request when both TIE bit and TDRE bit are "1".</p>

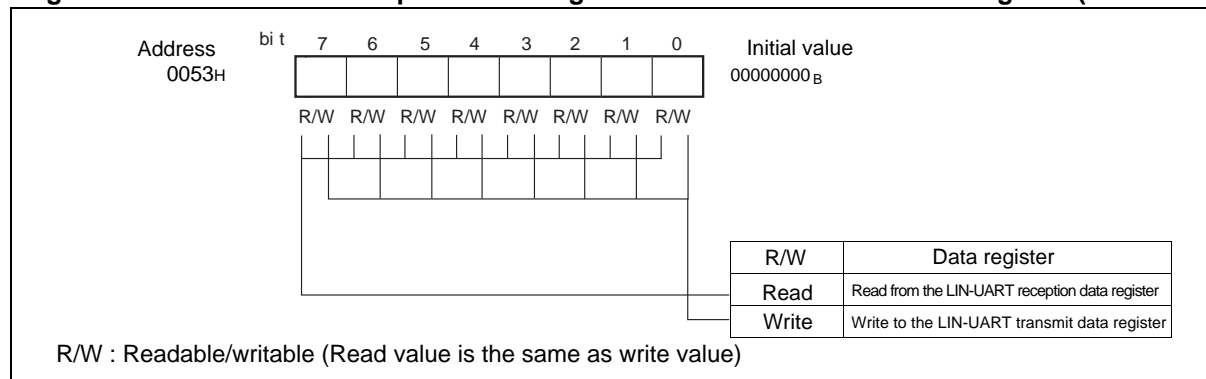
## 22.4.4 LIN-UART Reception Data Register/LIN-UART Transmit Data Register (RDR/TDR)

The LIN-UART reception and LIN-UART transmit data registers are located at the same address. If read, they work as the reception data register; if written, they work as the transmit data register.

### ■ LIN-UART Reception Data Register (RDR/TDR)

Figure 22.4-5 shows the LIN-UART reception data register/LIN-UART transmit data register.

**Figure 22.4-5 LIN-UART Reception Data Register/LIN-UART Transmit Data Register (RDR/TDR)**



### ■ LIN-UART Reception Data Register (RDR)

The LIN-UART reception data register (RDR) is the data buffer register for the serial data reception.

Serial data signal transmitted to the serial input pin (SIN pin) is converted via a shift register and stored in the LIN-UART reception data register (RDR).

If the data length is 7 bits, the upper 1 bit (RDR:D7) is "0".

The reception data full flag bit (SSR:RDRF) is set to "1" when received data is stored into the LIN-UART reception data register (RDR). If the reception interrupt is enabled (SSR:RIE = 1), a reception interrupt request is generated.

The LIN-UART reception data register (RDR) should be read when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is automatically cleared to "0" by reading the LIN-UART reception data register (RDR). Also, the reception interrupt is cleared when the reception interrupt is enabled and no error occurs.

When the reception error occurs (any of SSR:PE, ORE, or FRE is "1"), the data in the LIN-UART reception data register (RDR) is invalid.

### ■ LIN-UART Transmit Data Register (TDR)

The LIN-UART transmit data register (TDR) is the data buffer register for the serial data transmission.

If the data to be transmitted is written to the LIN-UART transmit data register (TDR) when transmission is enabled (SCR:TXE = 1), the transmit data is transferred to the transmission shift register, converted to serial data, and output from the serial data output pin (SOT pin).

If the data length is 7 bits, the data in the upper 1 bit (TDR:D7) is invalid.

The transmit data empty flag (SSR:TDRE) is cleared to "0" when a transmit data is written to the LIN-UART transmit data register (TDR).

The transmit data empty flag (SSR:TDRE) is set to "1" after the data is transferred to the transmission shift register and the transmission starts.

If the transmit data empty flag (SSR:TDRE) is "1", the next transmit data can be written. If the transmit interrupt is enabled, a transmit interrupt is generated. The next transmit data should be written by generating the transmit interrupt, or when the transmit data empty flag (SSR:TDRE) is "1".

---

**Note:**

The LIN-UART transmit data register is a write-only register; the reception data register is a read-only register. Since both registers are located at the same address, the write value and read value are different. Thus, the instructions to operate the read-modify-write (RMW) instruction, such as the INC/DEC instruction, cannot be used.

---

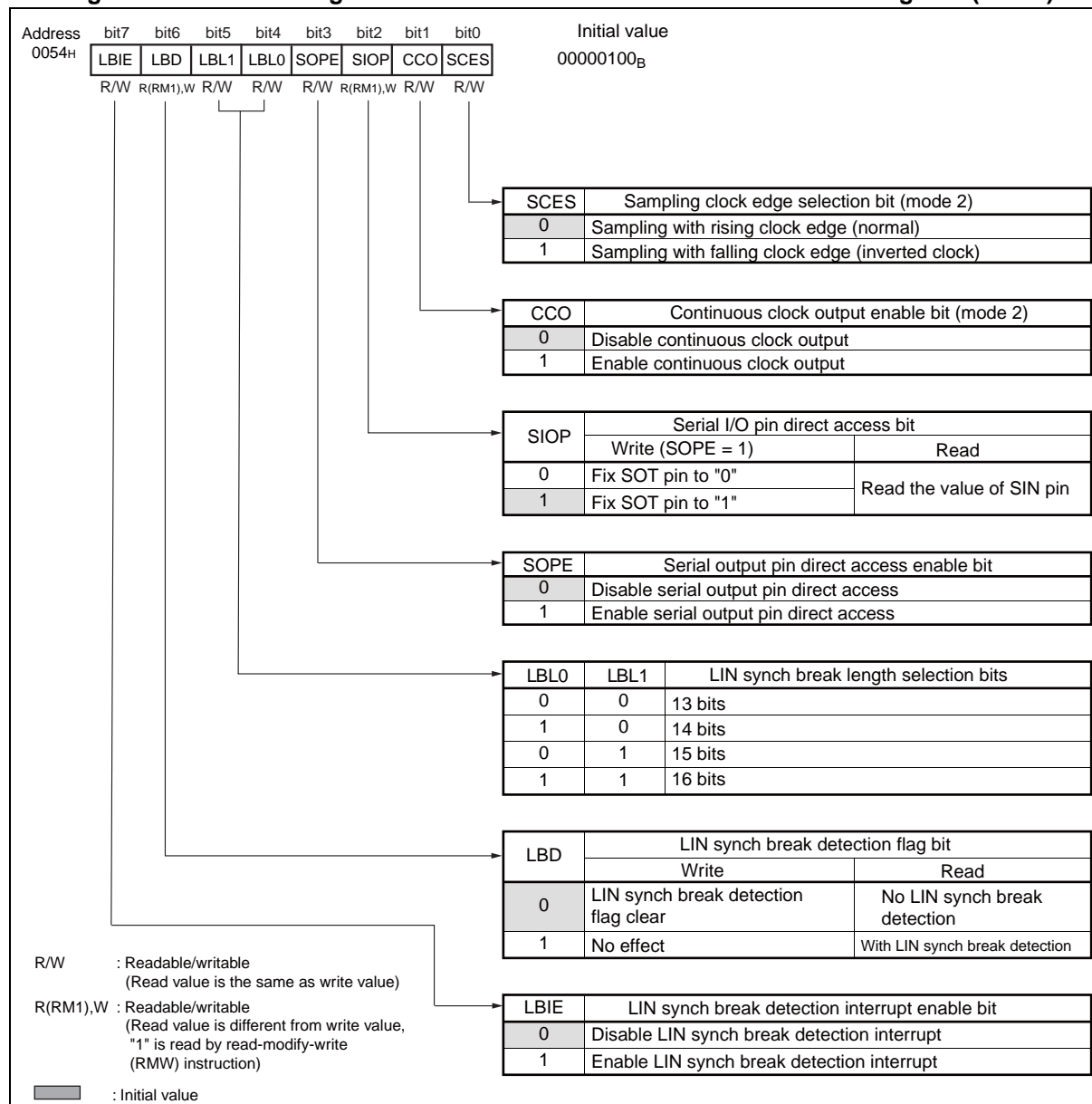
## 22.4.5 LIN-UART Extended Status Control Register (ESCR)

The LIN-UART extended status control register (ESCR) has the settings for enabling/disabling LIN synch break interrupt, LIN synch break length selection, LIN synch break detection, direct access to the SIN and SOT pins, continuous clock output in LIN-UART synchronous clock mode and sampling clock edge.

### ■ Bit Configuration of LIN-UART Extended Status Control Register (ESCR)

Figure 22.4-6 shows the bit configuration of the LIN-UART extended status control register (ESCR). Table 22.4-4 lists the function of each bit in LIN-UART extended status control register (ESCR).

Figure 22.4-6 Bit Configuration of LIN-UART Extended Status Control Register (ESCR)





**Table 22.4-4 Functions of Each Bit in LIN-UART Extended Status Control Register (ESCR)**

Bit name		Function
bit7	LBIE: LIN synch break detection interrupt enable bit	This bit enables or disables LIN synch break detection interrupts. An interrupt is generated when the LIN synch break detection flag (LBD) is "1" and the interrupt is enabled (LBIE = 1). This bit is fixed to "0" in mode 1 and mode 2.
bit6	LBD: LIN synch break detection flag bit	Detect LIN synch break. This bit is set to "1" when the LIN synch break is detected in operation mode 3 (the serial input is "0" when bit width is 11 bits or more). Also, writing "0" clears the LBD bit and the interrupt. Although the bit is always read as "1" when the read-modify-write (RMW) instruction is executed, this does not indicate that a LIN synch break detected. Note: To detect a LIN synch break, enable the LIN synch break detection interrupt (LBIE = 1), and then disable the reception (SCR:RXE = 0).
bit5, bit4	LBL1/LBL0: LIN synch break length selection bits	These bits specify the bit length for the LIN synch break generation time. The LIN synch break length for reception is always 11 bits.
bit3	SOPE: Serial output pin direct access enable bit*	Enable or disable direct writing to the SOT pin. Setting this bit to "1" when serial data output is enabled (SMR:SOE = 1) enables direct writing to the SOT pin.*
bit2	SIOP: Serial I/O pin direct access bit*	Control direct access to the serial I/O pin. Normal read instruction always returns the value of the SIN pin. When direct access to the serial output pin data is enabled (SOPE = 1), the value written to this bit reflects the SOT pin.* Note: The bit operation instruction returns the bit value of the SOT pin in the read cycle.
bit1	CCO: Continuous clock output enable bit	Enable or disable continuous serial clock output from the SCK pin. Setting this bit to "1" with sending side of serial clock in operation mode 2 (synchronous) enables the continuous serial clock output from the SCK pin if the pin is set as a clock output. Note: When the CCO bit is "1", the SSM bit in the ECCR should be "1".
bit0	SCES: Sampling clock edge select bit	Select a sampling edge. Setting the SCES to "1" in reception side of serial clock in operation mode 2 (synchronous) switches the sampling edge from the rising edge to the falling edge. When the SCK pin is set as the clock output with sending side of serial clock (ECCR:MS = 0) in operation mode 2, the internal serial clock and the output clock signal are inverted. This bit should be "0" in operation modes 0, 1, and 3.

**\*: Interaction between SOPE and SIOP**

SOPE	SIOP	Write to SIOP	Read from SIOP
0	R/W	No effect (however, the write value is retained)	Return the SIN value
1	R/W	Write "0" or "1" to SOT	Return the SIN value
1	RMW	Read the SOT value, write "0" or "1"	

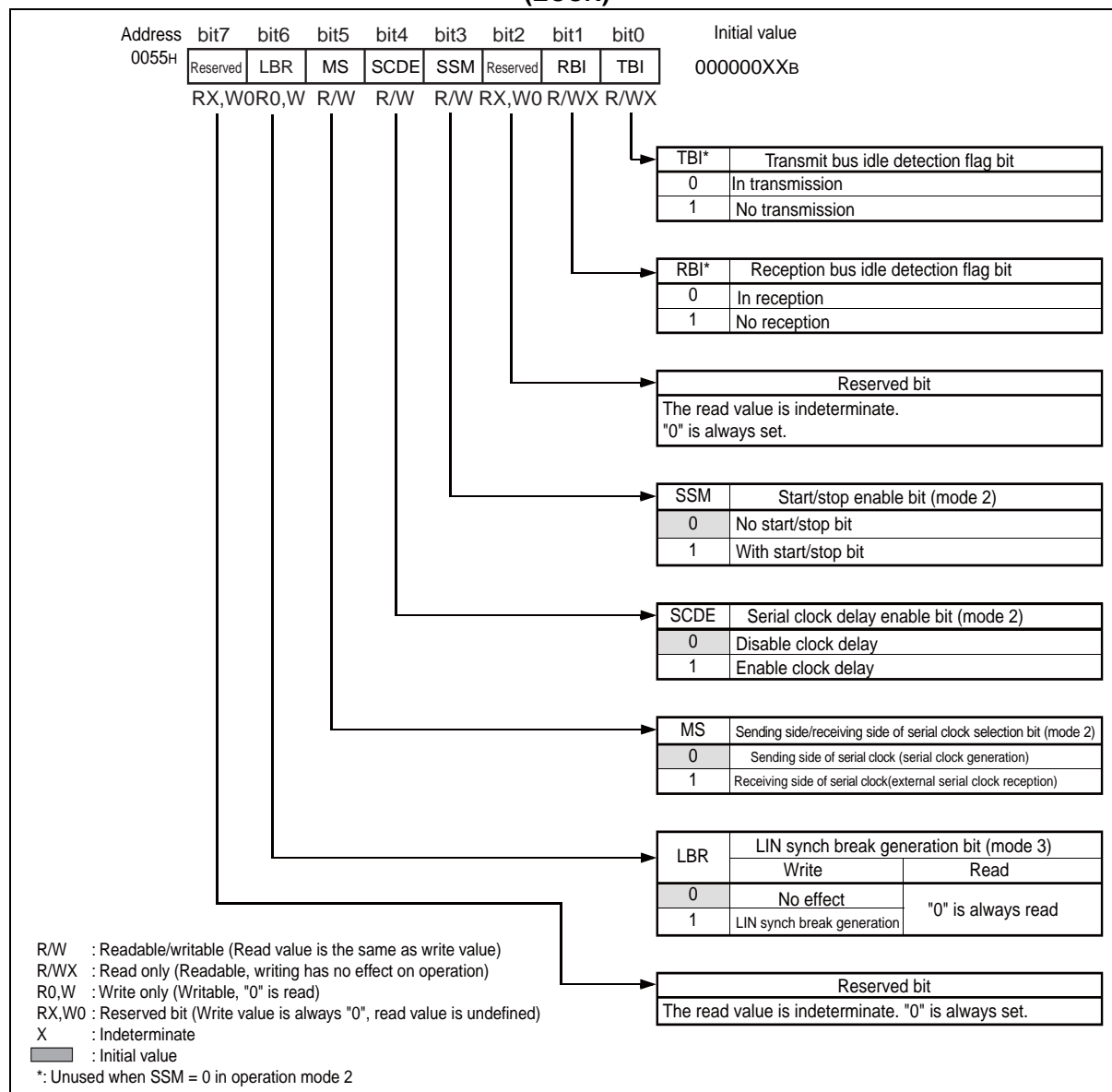
## 22.4.6 LIN-UART Extended Communication Control Register (ECCR)

The LIN-UART extended communication control register (ECCR) is used for the bus idle detection, the synchronous clock setting, and the LIN Synch break generation.

### ■ Bit Configuration of LIN-UART Extended Communication Control Register (ECCR)

Figure 22.4-7 shows the bit configuration of the LIN-UART extended communication control register (ECCR). Table 22.4-5 lists the function of each bit in the LIN-UART extended communication control register (ECCR).

**Figure 22.4-7 Bit Configuration of LIN-UART Extended Communication Control Register (ECCR)**



**Table 22.4-5 Functions of Each Bit in LIN-UART Extended Communication Control Register (ECCR)**

Bit name		Function
bit7	Reserved bit	The read value is indeterminate. "0" is always set.
bit6	LBR: LIN Synch break generation bit	Setting this bit to "1" in mode 3 generates a LIN synch break which has the length specified by LBL0/1 in the ESCR. This bit should be "0" in mode 0, 1, and 2.
bit5	MS: Sending side/receiving side of serial clock select bit	Select sending side/receiving side of serial clock in mode 2. When sending side "0" is selected, generate a synchronous clock. When receiving side "1" is selected, receive an external serial clock. This bit is fixed to "0" in modes 0, 1, and 3. Modify this bit only when the SCR:TXE bit is "0". Note: When receiving side of serial clock is selected, the clock source must be set as an external clock and the external clock input must be enabled (SMR:SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	Setting the SCDE bit to "1" at sending side of serial clock in mode 2 outputs a delayed serial clock as shown in Figure 22.7-5. This bit is effective in serial peripheral interface. This bit is fixed to "0" in modes 0, 1, and 3.
bit3	SSM: Start/stop bit mode enable bit	Add the start/stop bit to the synchronous data format when this bit is set to "1" in mode 2. This bit is fixed to "0" in modes 0, 1, and 3.
bit2	Reserved bit	The read value is indeterminate. "0" is always set.
bit1	RBI: Reception bus idle detection flag bit	When the SIN pin is "H" level and reception is not performed, this bit is "1". Do not use this bit when SSM = 0 in operation mode 2.
bit0	TBI: Transmit bus idle detection flag bit	This bit is "1" when there is no transmission on the SOT pin. Do not use this bit when SSM = 0 in operation mode 2.

## MB95110B/M Series

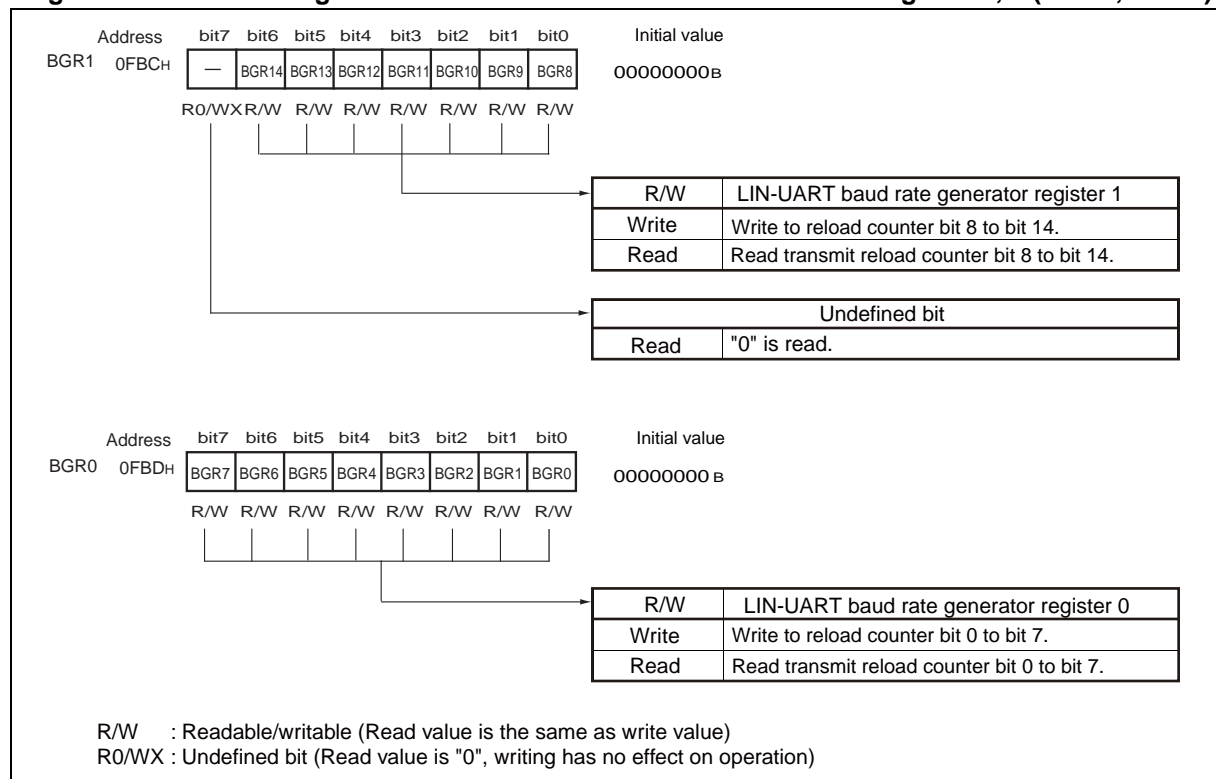
### 22.4.7 LIN-UART Baud Rate Generator Register 1, 0 1, 0 (BGR1, BGR0)

The LIN-UART baud rate generator register 1, 0 (BGR1, BGR0) sets the division ratio of the serial clock. Also, the count value in the transmit reload counter is read from this generator.

#### ■ Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0)

Figure 22.4-8 shows the bit configuration of LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

**Figure 22.4-8 Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0)**



The LIN-UART baud rate generator register sets the division ratio of the serial clock.

BGR1 is associated with the upper bits; BGR0 is associated with the lower bits. The reload value of the counter can be written and the transmit reload counter value can be read from them. Byte/word access is also possible.

Writing a reload value to the LIN-UART baud rate generator registers causes the reload counter to start counting.

Note:

Write to this register when LIN-UART stops.

## 22.5 Interrupt of LIN-UART

The LIN-UART has reception interrupts and transmit interrupts, which are generated by following factor and have the assigned interrupt number and interrupt vector. Also, it has the LIN synch field edge detection interrupt function using the 8/16-bit compound timer interrupt.

- **Reception interrupt**

When the received data is set in the reception data register (RDR), or when a reception error occurs. Also, when a LIN synch break is detected.

- **Transmit interrupt**

When the transmit data is transferred from the transmit data register (TDR) to the transmission shift register, and the transmission starts.

### ■ Reception Interrupt

Table 22.5-1 shows the interrupt control bits and interrupt factors of reception interrupts.

**Table 22.5-1 Interrupt Control Bits and Interrupt Factors of Reception Interrupts**

Interrupt request flag bit	Flag register	Operation mode				Interrupt source	Interrupt factor enable bit	Clearing of interrupt request flag
		0	1	2	3			
RDRF	SSR	○	○	○	○	Write received data to RDR	SSR:RIE	Read received data
ORE	SSR	○	○	○	○	Overrun error		Write "1" to reception error flag clear bit (SCR:CRE)
FRE	SSR	○	○	△	○	Framing error		
PE	SSR	○	×	△	×	Parity error		
LBD	ESCR	×	×	×	○	LIN synch break detection	ESCR:LBIE	Write "0" to ESCR:LBD

○: Used bit

×: Unused bit

△: Only ECCR:SSM = 1 available

#### ● Reception interrupt

Each flag bit in the LIN-UART serial status register (SSR) is set to "1" when any of following operation occurs in reception mode:

##### Data reception completed

When the reception data is transferred from the serial input shift register to the LIN-UART reception data register (RDR) (RDRF = 1)

##### Overrun error

When the following serial data is received when RDRF = 1 and the RDR is not read by the CPU (ORE = 1)

##### Framing error

When the stop bit reception error occurs (FRE = 1)

##### Parity error

When the parity detection error occurs (PE = 1)

A reception interrupt request is generated if the reception interrupt is enabled ( $SSR:RIE = 1$ ) when any of the above flag bits is "1".

RDRF flag is automatically cleared to "0" by reading the LIN-UART reception data register (RDR). All of error flags are cleared to "0" by writing "1" to the reception error flag clear bit (CRE) in the LIN-UART serial control register (SCR).

---

**Note:**

For the CRE bit, disable the reception operation ( $RXE=0$ ) first, and then clear the reception error flags. If an error flag is cleared before the reception operation is disabled, the reception will be aborted at that time, and resume later. This may result in a data reception error.

---

● **LIN synch break interrupts**

Works for LIN slave operation in operation mode 3.

The LIN synch break detection flag bit (LBD) in the LIN-UART extended status control register (ESCR) is set to "1" when the internal data bus (serial input) is "0" for 11 bits or longer. The LIN synch break interrupt and the LBD flag are cleared by writing "0" to the LBD flag. The LBD flag must be cleared before the 8/16-bit compound timer interrupt is generated in the LIN synch field.

To detect a LIN synch break, the reception must be disabled ( $SCR:RXE = 0$ ).

■ **Transmit Interrupts**

Table 22.5-2 shows the interrupt control bits and interrupt factors of transmit interrupts.

**Table 22.5-2 Interrupt Control Bits and Interrupt Factors of Transmit Interrupts**

Interrupt request flag bit	Flag register	Operation mode				Interrupt factor	Interrupt request enable bit	Clearing of interrupt request flag
		0	1	2	3			
TDRE	SSR	○	○	○	○	Transmit register is empty	SSR:TIE	Write transmit data

○: Used bit

● **Transmit interrupt**

The transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is set to "1" when the transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmission shift register, and the transmission starts. If the transmit interrupt is enabled ( $SSR:TIE = 1$ ) in this case, a transmit interrupt request is generated.

---

**Note:**

Since the initial value of TDRE is "1", an interrupt is generated immediately after the TIE bit is set to "1" after hardware/software reset. Also, the TDRE is cleared only by writing data to the LIN-UART transmit data register (TDR).

---

## ■ LIN Synch Field Edge Detection Interrupt (8/16-bit Compound Timer Interrupt)

Table 22.5-3 shows the interrupt control bits and interrupt factors of the LIN synch field edge detection interrupt.

**Table 22.5-3 Interrupt Control Bits and Interrupt Factors of LIN Synch Field Edge Detection Interrupt**

Interrupt request flag bit	Flag register	Operation mode				Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
		0	1	2	3			
IR	T00CR1	×	×	×	○	First falling edge of the LIN synch field	T00CR1:IE	Write "0" to T00CR1:IR
IR	T00CR1	×	×	×	○	Fifth falling edge of the LIN synch field		

○: Used bit  
X: Unused bit

### ● LIN synch field edge detection interrupt (8/16-bit compound timer interrupt)

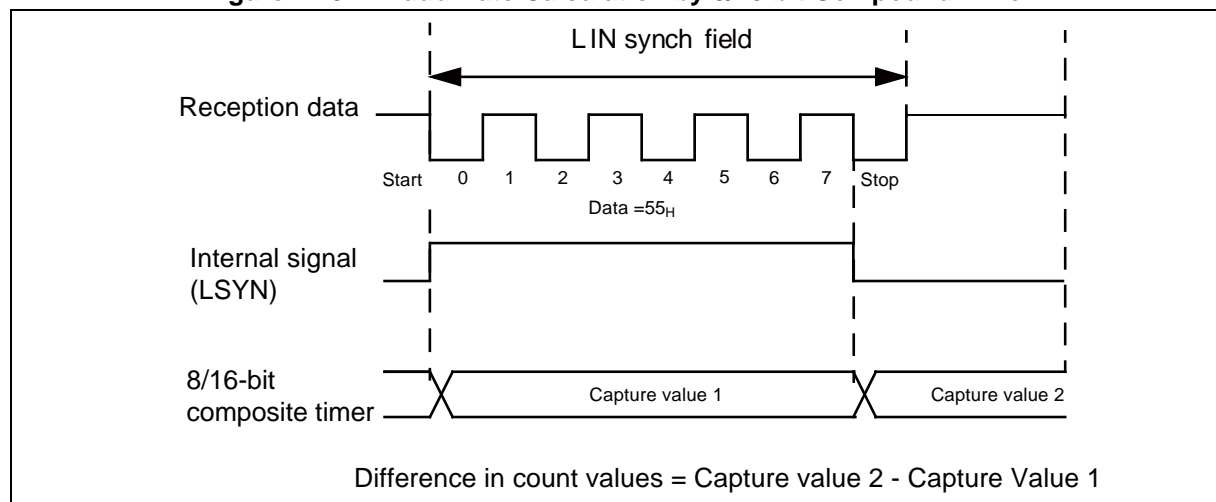
Works for LIN slave operation in operation mode 3.

After a LIN synch break is detected, the internal signal (LSYN) is set to "1" at the first falling edge of the LIN synch field, and set to "0" after the fifth falling edge. When the 8/16-bit compound timer is configured to be input the internal signal and to detect both edges, a 8/16-bit compound timer interrupt is generated if enabled.

The difference in the count values detected by the 8/16-bit compound timer (see Figure 22.5-1) corresponds to the 8 bits in the master serial clock. The new baud rate can be calculated from this value.

When a new baud rate is set, the rate become effective from the falling edge detection of the specified next start bit.

**Figure 22.5-1 Baud Rate Calculation by 8/16-bit Compound Timer**



## ■ Register and Vector Table Related to LIN-UART Interrupt

**Table 22.5-4 Register and Vector Table Related to LIN-UART Interrupt**

Interrupt sources	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
Reception	IRQ7	ILR1	L07	FFEC <sub>H</sub>	FFED <sub>H</sub>
Transmission	IRQ8	ILR2	L08	FFEA <sub>H</sub>	FFEB <sub>H</sub>



## 22.5.1 Reception Interrupt Generation and Flag Set Timing

Reception interrupts are a reception completion and an occurrence of a reception error.

### ■ Reception Interrupt Generation and Flag Set Timing

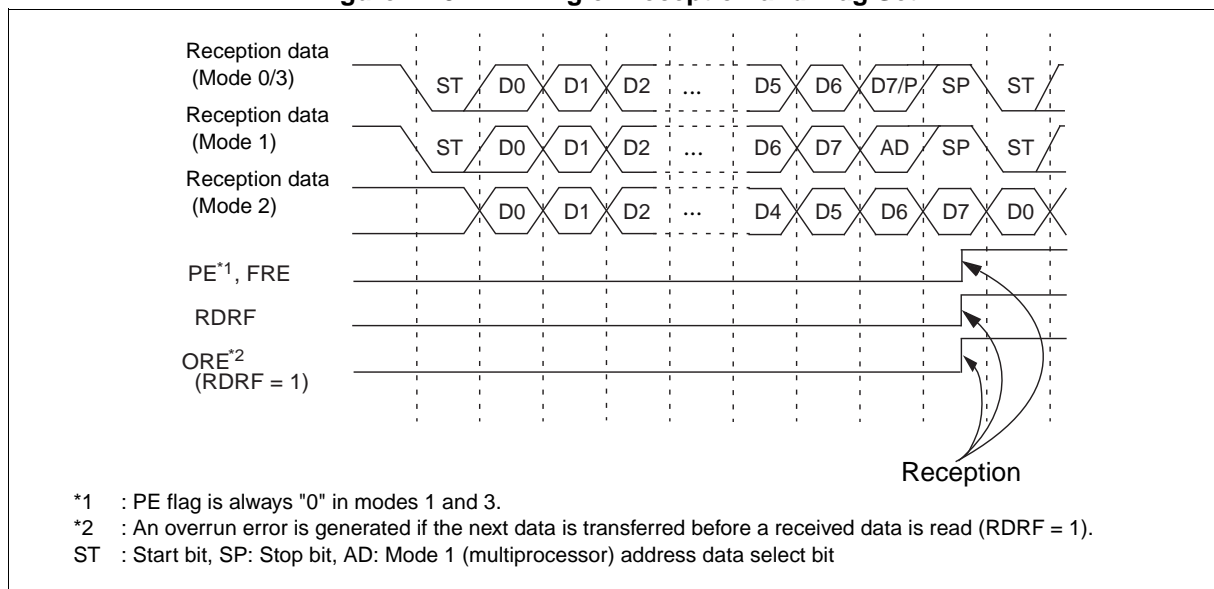
Received data is stored in the LIN-UART reception data register (RDR) when the first stop bit is detected in mode 0, 1, 2 (SSM = 1), 3, or when the last data bit is detected in mode 2 (SSM = 0). Each error flag is set when a reception is completed (SSR:RDRF = 1), or when a reception error occurs (SSR:PE, ORE, FRE = 1). If the reception interrupt is enabled (SSR:RIE = 1) at this time, a reception interrupt is generated.

Note:

When a reception error occurs in each mode, the data in the LIN-UART reception data register (RDR) is invalid.

Figure 22.5-2 shows the timing of reception and flag set.

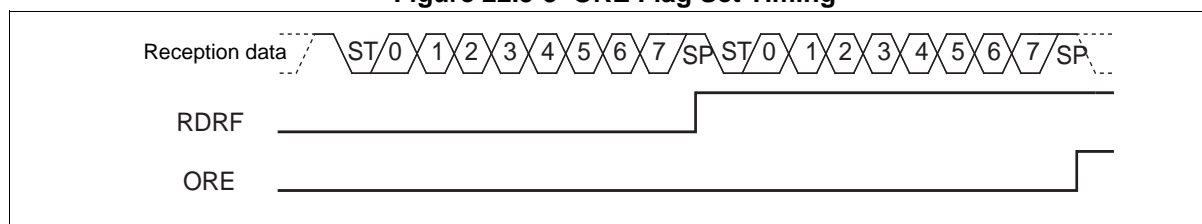
Figure 22.5-2 Timing of Reception and Flag Set



Note:

Figure 22.5-2 does not show all receptions in mode 0. It only shows examples for 7-bit data, parity (even parity or odd parity), 1 stop bit and 8-bit data, no parity, 1 stop bit.

**Figure 22.5-3 ORE Flag Set Timing**



## 22.5.2 Transmit Interrupt Generation and Flag Set Timing

Transmit interrupts are generated when the transmit data is transferred from the LIN-UART transmit data register (TDR) to the transmission shift register and then the transmission starts.

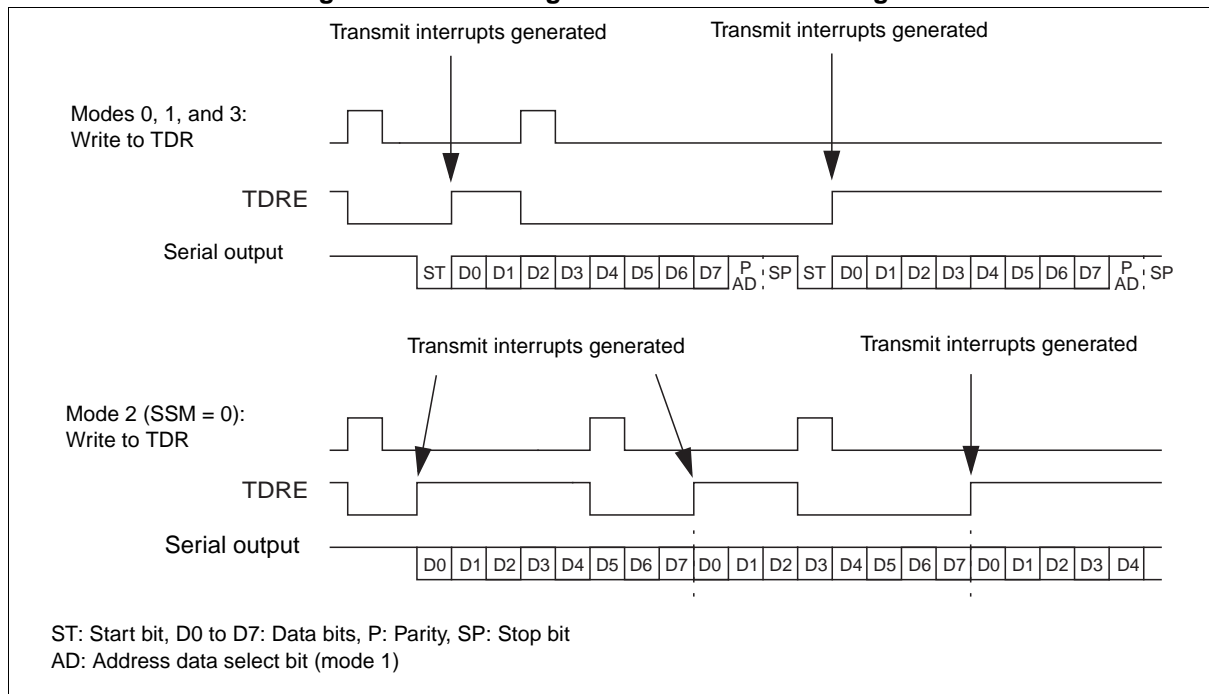
### ■ Transmit Interrupt Generation and Flag Set Timing

When the data written to the LIN-UART transmit data register (TDR) is transferred to the transmission shift register and then the transmission starts, the next data becomes to be writable (SSR:TDRE = 1). If the transmit interrupt is enabled (SSR:TIE = 1) at this time, a transmit interrupt is generated.

TDRE bit is a read-only bit and cleared to "0" only by writing data to the LIN-UART transmit data register (TDR).

Figure 22.5-4 shows the timing of the transmission and flag set.

Figure 22.5-4 Timing of Transmission and Flag Set



**Note:**

Figure 22.5-4 does not show all transmissions in mode 0. It only shows "8P1" (P= "even parity" or "odd parity").

No parity bit is transmitted in mode 3, or in mode 2 with SSM = 0.

**■ Transmit Interrupt Request Generation Timing**

When TDRE flag is set to "1" if the transmit interrupt is enabled (SSR:TIE = 1), a transmit interrupt is generated.

---

**Note:**

Since the TDRE bit is initially set to "1", a transmit interrupt is generated immediately after the transmit interrupt is enabled (SSR:TIE = 1). Be careful with the timing for enabling the transmit interrupt since the TDRE bit can be cleared only by writing new data to the LIN-UART transmit data register (TDR).

---

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

## 22.6 LIN-UART Baud Rate

---

One of the following can be selected for the LIN-UART input clock (send/receive clock source):

- Input a machine clock into a baud rate generator (reload counter)
  - Input an external clock into a baud rate generator (reload counter)
  - Use the external clock (SCK pin input clock) directly
- 

### ■ LIN-UART Baud Rate Selection

You can select one of the following three different baud rates. Figure 22.6-1 shows the LIN-UART baud rate selection circuit.

- Baud rate derived from the internal clock divided by the dedicated baud rate generator (reload counter)

Two internal reload counters are provided and assigned the transmit and reception serial clock respectively. The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock by the specified value.

It is used in asynchronous mode and in synchronous mode (sending side of serial clock).

To set the clock source, select the use of the internal clock and baud rate generator (SMR:EXT = 0, OTO = 0).

- Baud rate derived from the external clock divided by the dedicated baud rate generator (reload counter)

The external clock is used as the clock source for the reload counter.

The baud rate is selected by setting a 15-bit reload value in the LIN-UART baud rate generator register 1, 0 (BGR1, BGR0).

The reload counter divides the external clock by the specified value.

It is used in asynchronous mode.

To set the clock source, select the use of the external clock and baud rate generator (SMR:EXT = 1, OTO = 0).

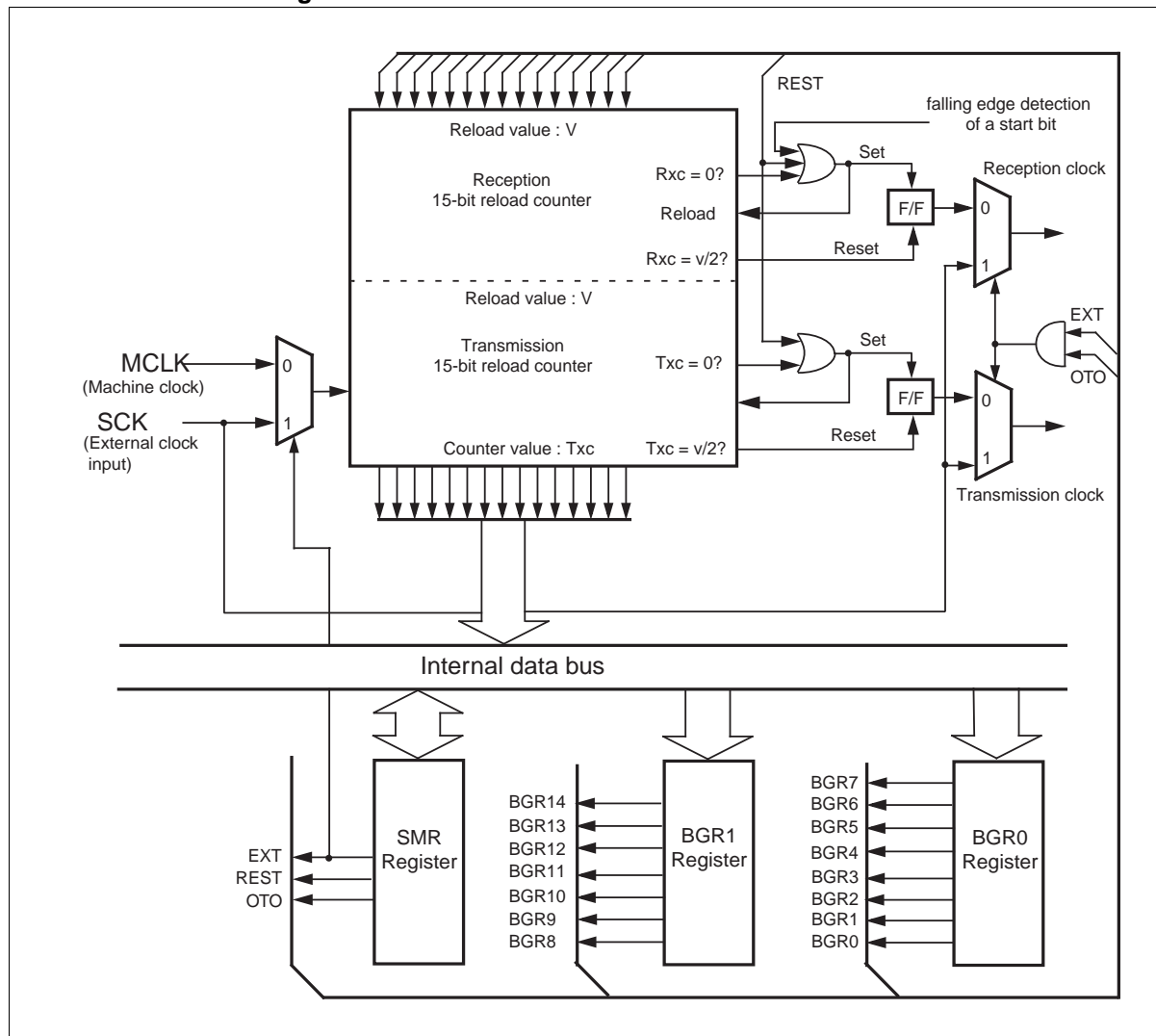
- Baud rate by the external clock (one-to-one mode)

The clock input from the clock input pin (SCK) of the LIN-UART is used as the baud rate (slave 2 operation (ECCR:MS = 1) in synchronous mode).

It is used in synchronous mode (receiving side of serial clock).

To set the clock source, select the external clock and its direct use (SMR:EXT = 1, OTO = 1).

Figure 22.6-1 LIN-UART Baud Rate Selection Circuit



## 22.6.1 Baud Rate Setting

---

**This section shows baud rate settings and the calculation result of serial clock frequencies.**

---

### ■ Baud Rate Calculation

The two 15-bit reload counters are set by the baud rate generator register 1, 0 (BGR1, BGR0).

The expressions for the baud rate are as follows.

Reload value:

$$v = \left( \frac{\text{MCLK}}{b} \right) - 1$$

v: Reload value, b: Baud rate, MCLK: Machine clock, or external clock frequency

#### Calculation example

Assuming that the machine clock is 10MHz, the internal clock is used, and the baud rate is set to 19200 bps:

Reload value:

$$v = \left( \frac{10 \times 10^6}{19200} \right) - 1 = 519.83... \cong 520$$

Thus, the actual baud rate can be calculated as follows.

$$b = \frac{\text{MCLK}}{(v + 1)} = \frac{10 \times 10^6}{521} = 19193.8579$$

---

#### Note:

The reload counter halts if the reload value is set to "0". Therefore, the least reload value should be "1".

For transmission/reception in asynchronous mode, the reload value must be at least "4" in order to determine the reception value by oversampling on five times.

---

**MB95110B/M Series****■ Reload Value and Baud Rate of Each Clock Speed**

Table 22.6-1 shows the reload value and baud rate.

**Table 22.6-1 Reload Value and Baud Rate**

Baud rate (bps)	8MHz (MCLK)		10MHz (MCLK)		16MHz (MCLK)		16.25MHz(MCLK)	
	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation	Reload value	Frequency deviation
2M	–	–	4	0	7	0	–	–
1M	7	0	9	0	15	0	–	–
500000	15	0	19	0	31	0	–	–
400800	–	–	–	–	–	–	–	–
250000	31	0	39	0	63	0	64	0
230400	–	–	–	–	68	- 0.64	–	–
153600	51	- 0.16	64	- 0.16	103	- 0.16	105	0.19
125000	63	0	79	0	127	0	129	0
115200	68	- 0.64	86	0.22	138	0.08	140	- 0.04
76800	103	0.16	129	0.16	207	- 0.16	211	0.19
57600	138	0.08	173	0.22	277	0.08	281	- 0.04
38400	207	0.16	259	0.16	416	0.08	422	- 0.04
28800	277	0.08	346	- 0.06	555	0.08	563	- 0.04
19200	416	0.08	520	0.03	832	- 0.04	845	- 0.04
10417	767	< 0.01	959	< 0.01	1535	< 0.01	1559	< 0.01
9600	832	- 0.04	1041	0.03	1666	0.02	1692	0.02
7200	1110	< 0.01	1388	< 0.01	2221	< 0.01	2256	< 0.01
4800	1666	0.02	2082	- 0.02	3332	< 0.01	3384	< 0.01
2400	3332	< 0.01	4166	< 0.01	6666	< 0.01	6770	< 0.01
1200	6666	< 0.01	8334	< 0.01	13332	< 0.01	13541	< 0.01
600	13332	< 0.01	16666	< 0.01	26666	< 0.01	27082	< 0.01
300	26666	< 0.01	–	–	53332	< 0.01	54166	< 0.01

The unit of frequency deviation (dev.) is %.MCLK indicates the machine clock.



### ■ External Clock

The external clock is selected by writing "1" to the EXT bit in the LIN-UART serial mode register (SMR). In the baud rate generator, the external clock can be used in the same way as the internal clock.

When slave operation is used in synchronous mode 2, select the one-to-one external clock input mode (SMR:OTO = 1). In this mode, the external clock input to SCK is input directly to the LIN-UART serial clock.

---

#### Note:

The external clock signal is synchronized with the internal clock (MCLK: machine clock) in the LIN-UART. Therefore, the signal is unstable because the external clock cannot be divided if its cycle is faster than half cycle of the internal clock.

Be sure not to set the cycle of the external clock is faster than half cycle of the internal clock.

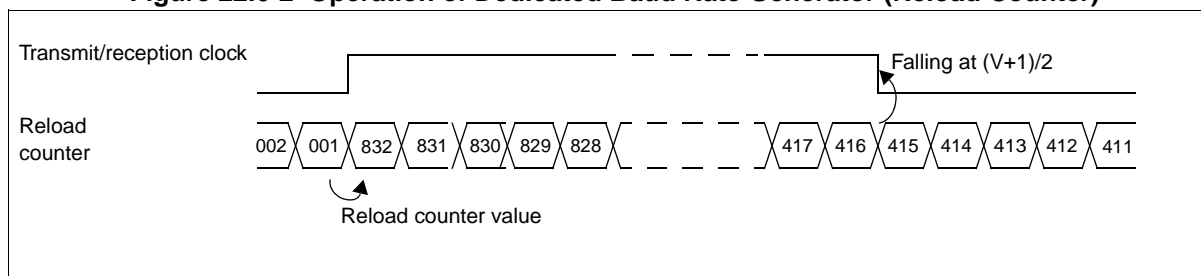
For the value of the SCK clock, see "Data Sheet".

---

## ■ Operation of Dedicated Baud Rate Generator (Reload Counter)

Figure 22.6-2 shows the operation of dedicated baud rate generator (reload counter).

**Figure 22.6-2 Operation of Dedicated Baud Rate Generator (Reload Counter)**



### Note:

The falling edge of the serial clock signal is generated after the reload value divided by 2 (  $(v+1)/2$  ) is counted.

## 22.6.2 Reload Counter

---

**This block is a 15-bit reload counter serving as a dedicated baud rate generator. It generates the transmit/reception clock from the external or internal clock.**

**The count value in the transmit reload counter is read from the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).**

---

### ■ Function of Reload Counter

There are two kinds of reload counters; transmit and reception. They work as the dedicated baud rate generator. The block consists of a 15-bit register for reload values; it generates the transmit/reception clock from the external or internal clock. The count value in the transmit reload counter is read from the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0).

#### ● Start counting

Writing a reload value to the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) causes the reload counter to start counting.

#### ● Restart

The reload counter restarts in the following conditions.

For both transmit/reception reload counter

- LIN-UART programmable reset (SMR:UPCL bit)
- Programmable restart (SMR:REST bit)

For reception reload counter

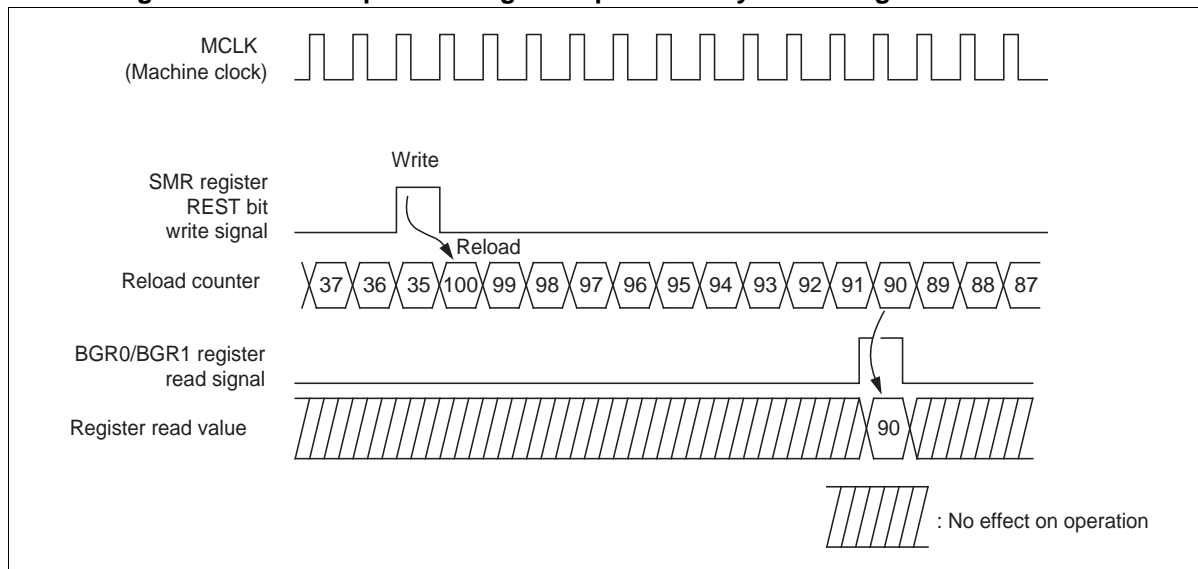
Start bit falling edge detection in asynchronous mode

#### ● Simple timer function

Two reload counters restart at the next clock cycle when the REST bit in the LIN-UART serial mode register (SMR) is set to "1".

This function enables the transmit reload counter to be used as a simple timer.

Figure 22.6-3 shows an example of using a simple timer by restarting the reload timer (when reload value is 100).

**Figure 22.6-3 Example of Using a Simple Timer by Restarting the Reload Timer**

The number of machine cycles "cyc" after restart in this example is obtained by the following expression.

$$\text{cyc} = v - c + 1 = 100 - 90 + 1 = 11$$

v: Reload value, c: Reload counter value

---

**Note:**

The reload counters also restart when the LIN-UART is reset by writing "1" to the SMR:UPCL bit.

---

**Automatic restart (reception reload counter only)**

The reception reload counter is restarted when the start bit falling edge is detected in asynchronous mode. This is the function to synchronize the reception shift register with the reception data.

● **Clear counter**

When a reset occurs, the reload values in the LIN-UART baud rate generator registers 1, 0 (BGR1, BGR0) and the reload counter are cleared to "00<sub>H</sub>", and the reload counter halts.

Although the counter value is temporarily cleared to "00<sub>H</sub>" by the LIN-UART reset (writing "1" to SMR:UPCL), the reload counter restarts since the reload value is retained.

The counter value is not cleared to "00<sub>H</sub>" by the restart setting (writing "1" to SMR:REST), and the reload counter restarts.

## 22.7 Operations and Setup Procedure Example of LIN-UART

LIN-UART operates in mode 0, 2 for bi-directional serial communication, in mode 1 for master/slave communication, and in mode 3 for LIN master/slave communication.

### ■ Operation of LIN-UART

#### ● Operation mode

The LIN-UART has four operation modes (0 to 3), allowing the connections between CPUs and the data transfer methods to be selected as listed in Table 22.7-1.

**Table 22.7-1 LIN-UART Operation Modes**

Operation mode		Data length		Synchronous method	Stop bit length	Data bit format
		No parity	With parity			
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits	LSB first MSB first
1	Multiprocessor mode	7 bits or 8 bits + 1*	—	Asynchronous		
2	Normal mode	8 bits		Synchronous	None, 1 bit, 2 bits	LSB first
3	LIN mode	8 bits	—	Asynchronous	1 bit	

—: Setting disabled

\*: "+1" is the address/data selection bit (AD) used for communication control in multiprocessor mode.

The MD0 and MD1 bits in the LIN-UART serial mode register (SMR) are used to select the following LIN-UART operation modes.

**Table 22.7-2 LIN-UART Operation Modes**

MD1	MD0	Mode	Type
0	0	0	Asynchronous (Normal mode)
0	1	1	Asynchronous (Multiprocessor mode)
1	0	2	Synchronous (Normal mode)
1	1	3	Asynchronous (LIN mode)

#### Notes:

- Both master and slave operation are supported in a system with master/slave connection in mode 1.
- In mode 3, the communication format is fixed to 8-bit data, no parity, 1 stop bit, LSB-first.
- If the mode is changed, all transmissions and receptions are canceled, and the LIN-UART waits for the next operation.

### ■ Inter-CPU Connection Method

You can select either external clock one-to-one connection (normal mode) or master/slave connection (multiprocessor mode). In either methods, data length, parity setting, synchronization type must be the same between all CPUs and thus the operation mode must be selected as follows.

- One-to-one connection: Two CPUs must use the same method in either operation mode 0 or 2. Choose operation mode 0 in an asynchronous system and operation mode 2 in a synchronous system. Also, for the operation mode 2, set one CPU as sending side of serial clock and the other as the receiving side of serial clock.
- Master/slave connection: Select operation mode 1. Use the system as a master/slave system.

### ■ Synchronous Method

In asynchronous method, the reception clock is synchronized with the reception start bit falling edge. In synchronous method, the reception clock can be synchronized by the sending side of serial clock signal or the clock signal at operating as sending side of serial clock.

### ■ Signaling

NRZ (Non Return to Zero).

### ■ Enable Transmission/Reception

The LIN-UART uses the SCR:TXE bit and the SCR:RXE bit to control transmission and reception, respectively. To disable transmission or reception, set as follows.

- If the reception is in progress, wait until the reception completed, read the reception data register (RDR), and then disable the reception.
- If the transmission is in progress, wait until the transmission completed, and then disable the transmission.

### ■ Setup Procedure Example

LIN-UART is set in the following procedure:

#### ● Initial setting

- 1) Set the port input (DDR6).
- 2) Set the interrupt level (ILR1, ILR2).
- 3) Set the data format, enable transmission/reception (SCR).
- 4) The operation mode, baud rate selection, pin output enabled (SMR)
- 5) The baud rate generator 1, 0 (BGR1, BGR0)

## 22.7.1 Operation of Asynchronous Mode (Operation Mode 0, 1)

---

**When LIN-UART is used in operation mode 0 (normal mode) or operation mode 1 (multiprocessor mode), the transfer method is asynchronous.**

---

### ■ Asynchronous Mode Operation

#### ● Transmit/reception data format

Transmit/reception data always begins with a start bit ("L" level) followed by a specified data bits length and ends up with at least one stop bit ("H" level).

The bit transfer direction (LSB-first or MSB-first) is determined by the BDS bit in the LIN-UART serial status register (SSR). When a parity is used, the parity bit is always placed between the last data bit and the first stop bit.

In operation mode 0, select 7-bit or 8-bit for the data length. You can select whether or not to use a parity. Also, the stop bit length (1 or 2) can be selected.

In operation mode 1, a data length is 7-bit or 8-bit, the parity is not added, and the address/data bit is added. The stop bit length (1 or 2) can be selected.

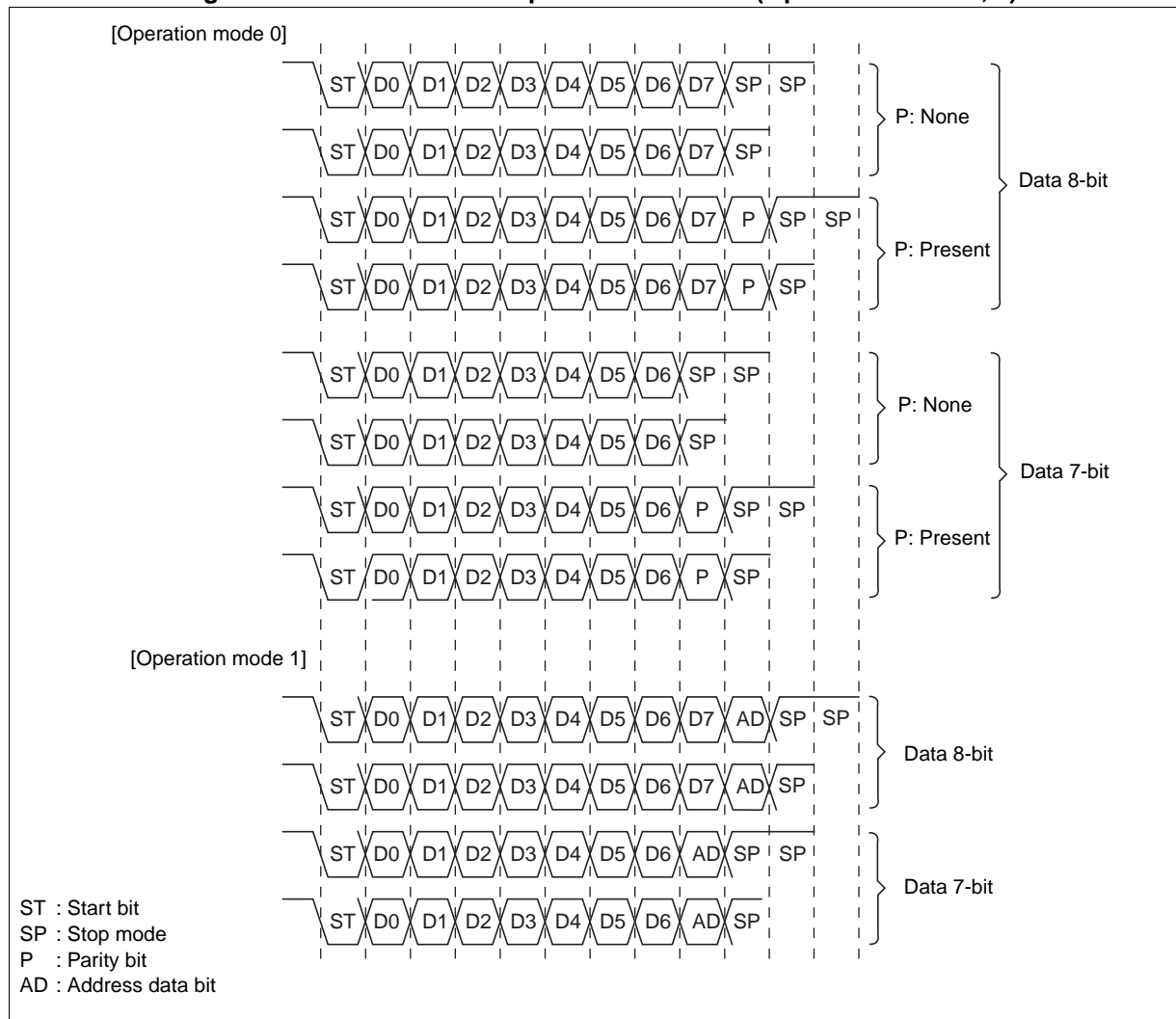
The bit length of transmit/reception frame is calculated as follows:

$$\text{Length} = 1 + d + p + s$$

(d = Number of data bits [7 or 8], p = parity [0 or 1],

s = Number of stop bits [1 or 2])

Figure 22.7-1 shows the transmit/reception data format (Operation Mode 0, 1).

**Figure 22.7-1 Transmit/Reception Data Format (Operation Mode 0, 1)****Note:**

When the BDS bit in the LIN-UART serial status register (SSR) is set to "1" (MSB-first), the bits are processed in the order of D7, D6, ... D1, D0 (P).



● Transmission

If the transmit data register empty flag bit (TDRE) in the LIN-UART serial status register (SSR) is "1", transmit data can be written into the LIN-UART transmit data register (TDR). Writing data sets the TDRE flag to "0". If transmission is enabled (SCR:TXE=1) at this time, the data is written to the transmit shift register and the transmission is started sequentially from the start bit in the next serial clock cycle.

If the transmit interrupt is enabled (TIE=1), the transmit data is transferred from the LIN-UART transmit data register (TDR) to transmit shift register, the TDRE flag is set to "1", and an interrupt occurs.

When the data length is set to 7-bit (CL=0), the bit7 in the TDR register is an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

---

Note:

Since the initial value of transmit data empty flag bit (SSR:TDRE) is "1", an interrupt is generated immediately when transmit interrupts are enabled (SSR:TIE =1).

---

● Reception

The reception is performed when reception is enabled (SCR:RXE =1). When the start bit is detected, one frame data is received according to the data format defined in the LIN-UART serial control register (SCR). If an error occurs, the error flag (SSR:PE, ORE, FRE) is set. After the reception of the one frame data is completed, the received data is transferred from the reception shift register to the LIN-UART reception data register (RDR), and the reception data register full flag bit (SSR:RDRF) is set to "1". If the reception interrupt request is enabled (SSR:RIE = 1) at this time, a reception interrupt request is output.

To read the received data, check the error flag status and read the received data from the LIN-UART reception data register (RDR) if the reception is normal. If a reception error occurs, perform error handlings.

When the received data is read, the reception data register full flag bit (SSR:RDRF) is cleared to "0".

When the data length is set to 7-bit (CL=0), the bit7 in the RDR register is an unused bit regardless of the transfer direction select bit (BDS) setting (LSB-first or MSB-first).

---

Note:

Data in the LIN-UART reception data register (RDR) becomes valid when the reception data register full flag bit (SSR:RDRF) is set to "1" and no error occurs (SSR:PE, ORE, FRE=0).

---

● Input clock

Internal or external clock is used. For the baud rate, select the baud rate generator (SMR:EXT = 0 or 1, OTO = 0).

● Stop bit and reception bus idle flag

You can select one or two stop bits at transmission. When 2-bit of the stop bit are selected, both of the stop bits are detected during reception.

When the first stop bit is detected, the reception data register full flag (SSR:RDRF) is set to "1". When no start bit is detected after that, the reception bus idle flag (ECCR:RBI) is set to "1", indicating that the reception is not performed.

● Error detection

In mode 0, parity, overrun, and framing errors can be detected.

In mode 1, overrun and framing errors can be detected. But, parity errors cannot be detected.

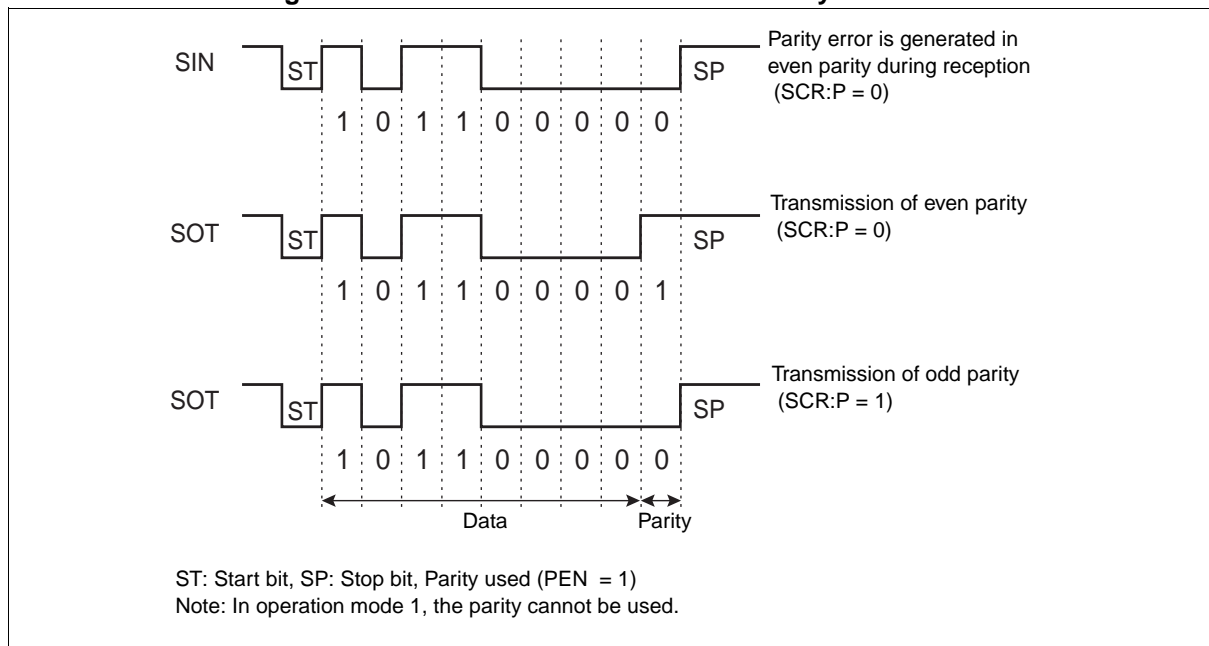
● Parity

You can specify whether or not to add (at transmission) and detect (at reception) a parity bit.

The parity enable bit (SCR:PEN) can be used whether or not to use a parity; the parity selection bit (SCR:P) can be used to select the odd or even parity.

In operation mode 1, the parity cannot be used.

**Figure 22.7-2 Transmission Data when Parity is Enabled**



● Data signaling

NRZ data format.

● Data transfer method

The data bit transfer method can be the LSB-first or MSB-first.

## 22.7.2 Operation of Synchronous Mode (Operation Mode 2)

When LIN-UART is used in operation mode 2 (normal mode), the transfer method is clock synchronous.

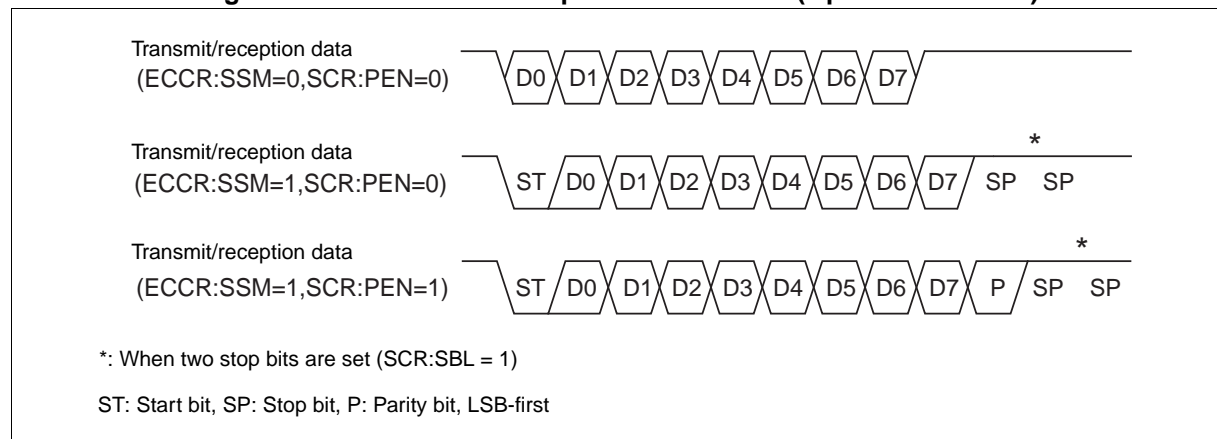
### ■ Operation of Synchronous Mode (Operation Mode 2)

#### ● Transmit/reception data format

In synchronous mode, you can transmit and receive 8-bit data and select whether or not to include the start bit and stop bit (ECCR:SSM). When the start/stop bit is included (ECCR:SSM = 1), you can select whether or not to include the parity bit (SCR:PEN).

Figure 22.7-3 shows the transmit/reception data format (operation mode 2).

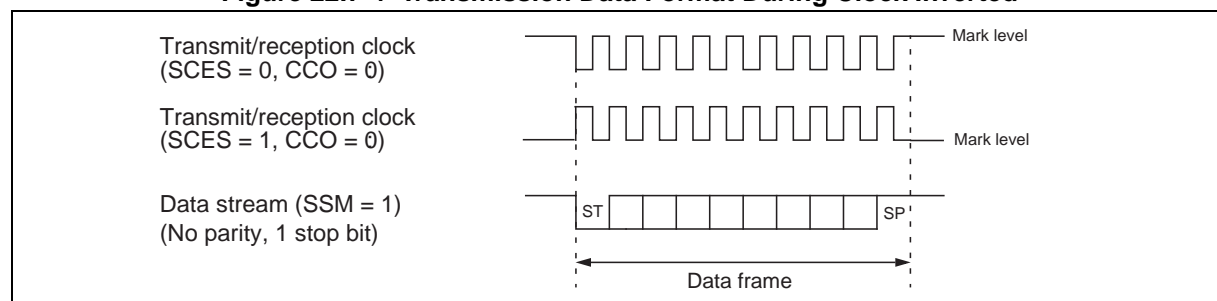
**Figure 22.7-3 Transmit/Reception Data Format (Operation Mode 2)**



#### ● Clock inversion function

When the SCES bit in the LIN-UART extended status control register (ESCR) is "1", the serial clock is inverted. In receiving side of serial clock, the LIN-UART samples data at the falling edge of the received serial clock. Note that, in sending side of serial clock, the mark level is set to "0" when the SCES bit is "1".

**Figure 22.7-4 Transmission Data Format During Clock Inverted**



### ● Start/stop bit

When the SSM bit in the LIN-UART extended communication control register (ECCR) is "1", the start and stop bits are added to the data format as in asynchronous mode.

### ● Clock supply

In clock synchronous mode (normal), the number of the transmit/reception bits must be equal to the number of the clock cycles. When the start/stop bit is enabled, the number of the added start/stop bits must be equal, as well.

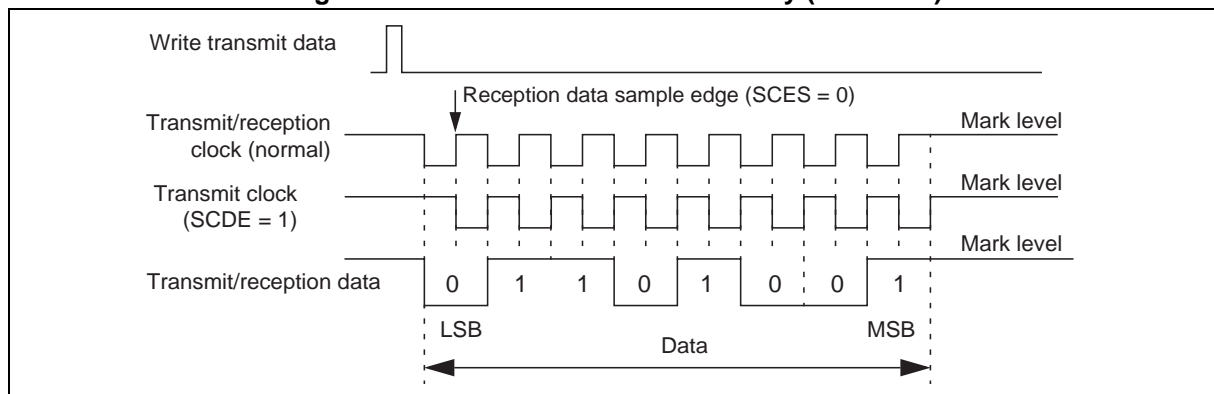
When the serial clock output is enabled (SMR: SCKE = 1) in sending side of serial clock (ECCR:MS = 0), a synchronous clock is output automatically at transmission/reception. When the serial clock output is disabled (SMR: SCKE = 0) in receiving side of serial clock (ECCR:MS = 1), the clock for each bit of transmit/reception data must be supplied from the outside.

The clock signal must remain at the mark level ("H") as long as it is irrelevant to transmission/reception.

### ● Clock delay

Setting the SCDE bit in the ECCR to "1", a delayed transmit clock is output as shown in Figure 22.7-5. This function is required when the receiving device samples data at the rising or falling edge of the clock.

**Figure 22.7-5 Transmission Clock Delay (SCDE = 1)**



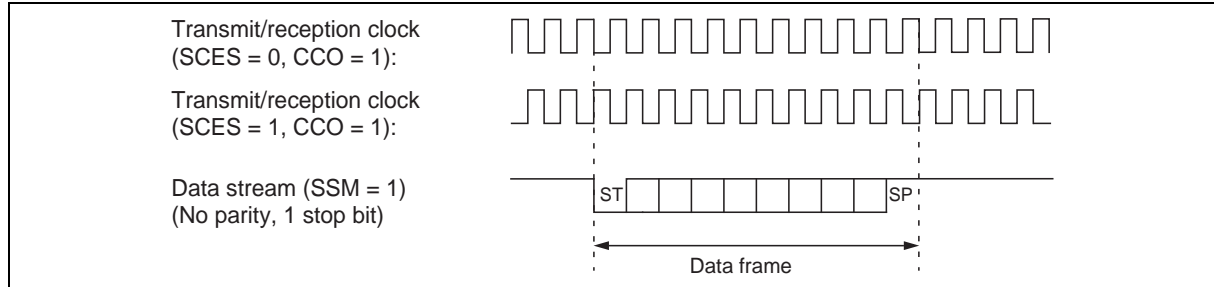
### ● Clock inversion

When the SCES bit in the LIN-UART extended status register (ESCR) is "1", the LIN-UART clock is inverted, and received data is sampled at the falling edge of the clock. At this time, the value of the serial data must be enabled at the timing of the clock falling edge.

- Continuous clock supply

When the CCO bit in the ESCR is "1", the serial clock output from the SCK pin is supplied in the sending side of serial clock continuously. In this mode, add the start/stop bit to the data format (SSM = 1) in order to identify the beginning and end of the data frame. Figure 22.7-6 shows the continuous clock supply (mode 2).

### Figure 22.7-6 Continuous Clock Supply (Mode 2)



- Error detection

When the start/stop bits are disabled (ECCR:SSM=0), only overrun errors are detected.

## ● Communication settings for synchronous mode

To communicate in synchronous mode, the following settings are required.

- LIN-UART baud rate generator register 1, 0 (BGR0, BGR1)
  - Set the dedicated baud rate reload counter to a required value.
- LIN-UART serial mode register (SMR)
  - MD1, MD0: "10<sub>B</sub>" (Mode 2)
  - SCKE: "1": Use the dedicated baud rate reload counter  
 "0": Input external clock
  - SOE: "1": Enable transmission/reception  
 : "0": Enable reception only
- LIN-UART serial control register (SCR)
  - RXE, TXE: Set either bit to "1".
  - AD : The value of this bit is disabled so that the address/data selection function cannot be used.
  - CL : This bit is set to 8 bits length automatically, and its value is disabled.
  - CRE : "1": Since the error flag is cleared, transmission/reception is stopped.
  - For SSM = 0:
    - PEN, P, SBL: Since not used, parity bit and stop bit are disabled.
  - For SSM = 1:
    - PEN : "1": Add/detect parity bit,                      "0": Not use parity bit
    - P : "1": Odd parity,   "0": Even parity
    - SBL : "1": Stop bit length 2,                               "0": Stop bit length 1

- LIN-UART serial status register (SSR)
  - BDS: "0": LSB first, "1": MSB-first
  - RIE: "1": Enable reception interrupt, "0": Disable reception interrupt
  - TIE: "1": Enable transmit interrupt, "0": Disable transmit interrupt
- LIN-UART extended communication control register (ECCR)
  - SSM: "0": Not use start/stop bit (normal),  
"1": Use start/stop bit (extended function),
  - MS: "0": Sending side of serial clock (serial clock output),  
"1": Receiving side of serial clock (input serial clock from sending side of serial clock)

---

**Note:**

To start communication, write data into the LIN-UART transmit data register (TDR).

To receive data, disable the serial output (SMR:SOE = 0), and then write dummy data into the TDR.

Enabling continuous clock and start/stop bit allows bi-directional communication as in asynchronous mode.

---

### 22.7.3 Operation of LIN function (Operation Mode 3)

In operation mode 3, the LIN-UART works as the LIN master and the LIN slave.  
In operation mode 3, the communication format is set to 8-bit data, no parity, stop bit1, LSB first.

#### ■ Asynchronous LIN Mode Operation

##### ● Operation as LIN master

In LIN mode, the master determines the baud rate for the entire bus, and the slave synchronizes to the master.

Writing "1" to the LBR bit in the LIN-UART extended communication control register (ECCR) outputs 13 to 16 bits at the "L" level from the SOT pin. This bit is the LIN synch break signifying the beginning of a LIN message.

The TDRE flag bit in the LIN-UART serial status register (SSR) is set to "0". After the break, it is set to "1" (initial value). If the TIE bit in SSR is "1" at this time, a transmit interrupt is output.

The length of the LIN Synch break transmitted is set by the LBL0/LBL1 bits in ESCR as in the following table.

**Table 22.7-3 LIN Break Length**

LBL0	LBL1	Break length
0	0	13 bits
1	0	14 bits
0	1	15 bits
1	1	16 bits

Synch field is transmitted as byte data 55<sub>H</sub> following the LIN break. To prevent generation of a transmit interrupt, 55<sub>H</sub> can be written to the TDR after the LBR bit in ECCR is set to "1" even if the TDRE flag is "0".

##### ● Operation as LIN slave

In LIN slave mode, the LIN-UART must synchronize to the baud rate for the master. The LIN-UART generates a reception interrupt when LIN break interrupt is enabled (LBIE = 1) even though reception is disabled (RXE = 0). The LBD bit in the ESCR is set to "1" at this time.

Writing "0" to the LBD bit clears the reception interrupt request flag.

For calculation of the baud rate, the following example shows the operation of the LIN-UART. When the LIN-UART detects the first falling edge of Synch field, set an internal signal, which is input to the 8/16-bit compound timer, to "H", and then start the timer. The internal signal should be "L" at the fifth falling edge. The 8/16-bit compound timer must be set to the input capture mode. Also, the 8/16-bit compound timer interrupts must be enabled and set for the detection at both edges. The time for which the input signal to the 8/16-bit compound timer is "1" becomes the value obtained by multiplying the baud rate by 8.

The baud rate setting value is calculated by the following expressions.

When the counter of the 8/16-bit compound timer is not overflowing  
: BGR value = (b - a)/8 - 1

When the counter of the 8/16-bit compound timer is overflowing

$$: \text{BGR value} = (\text{max} + \text{b} - \text{a}) / 8 - 1$$

max: Maximum value of free-run timer

a: TH0 data register value after the first interrupt

b: TH0 data register value after the second interrupt

**Note:**

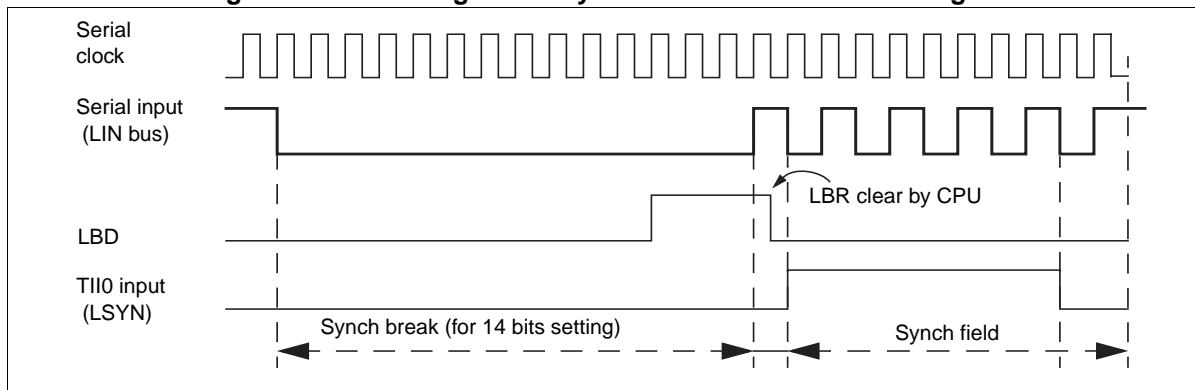
Do not set the baud rate if the new BGR value calculated based on Synch field as above in LIN slave mode involves an error over  $\pm 15\%$ .

For the operations of the input capture function on the 8/16-bit compound timer, see Section "15.13 Operating Description of Input Capture Function".

● LIN synch break detection interrupt and flag

The LIN break detection (LBD) flag in ESCR is set to "1" when the LIN synch break is detected in slave mode. When the LIN break interrupt is enabled (LBIE = 1), an interrupt is generated.

**Figure 22.7-7 Timing of LIN Synch Break Detection and Flag Set**



The above diagram shows the timing of the LIN synch break detection and flag.

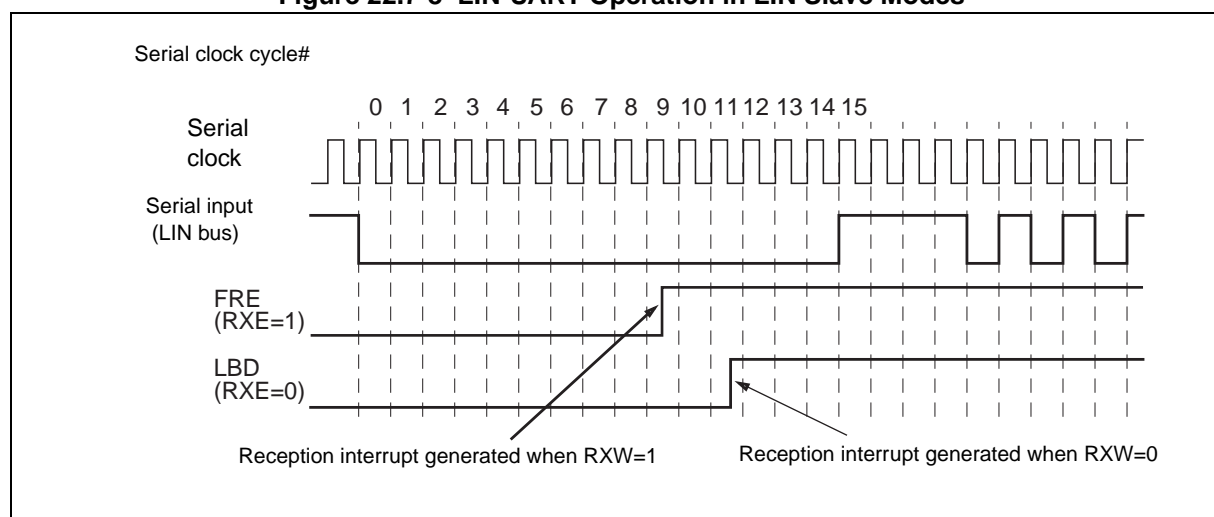
Since the data framing error (FRE) flag bit in SSR generates a reception interrupt two bits earlier than a LIN break interrupt (for communication format is 8-bit data, no parity, "1" stop bit.), set the RXE to "0" when a LIN break is used.

The LIN synch break detection works only in operation mode 3.

Figure 22.7-8 shows the LIN-UART operation in LIN slave modes.

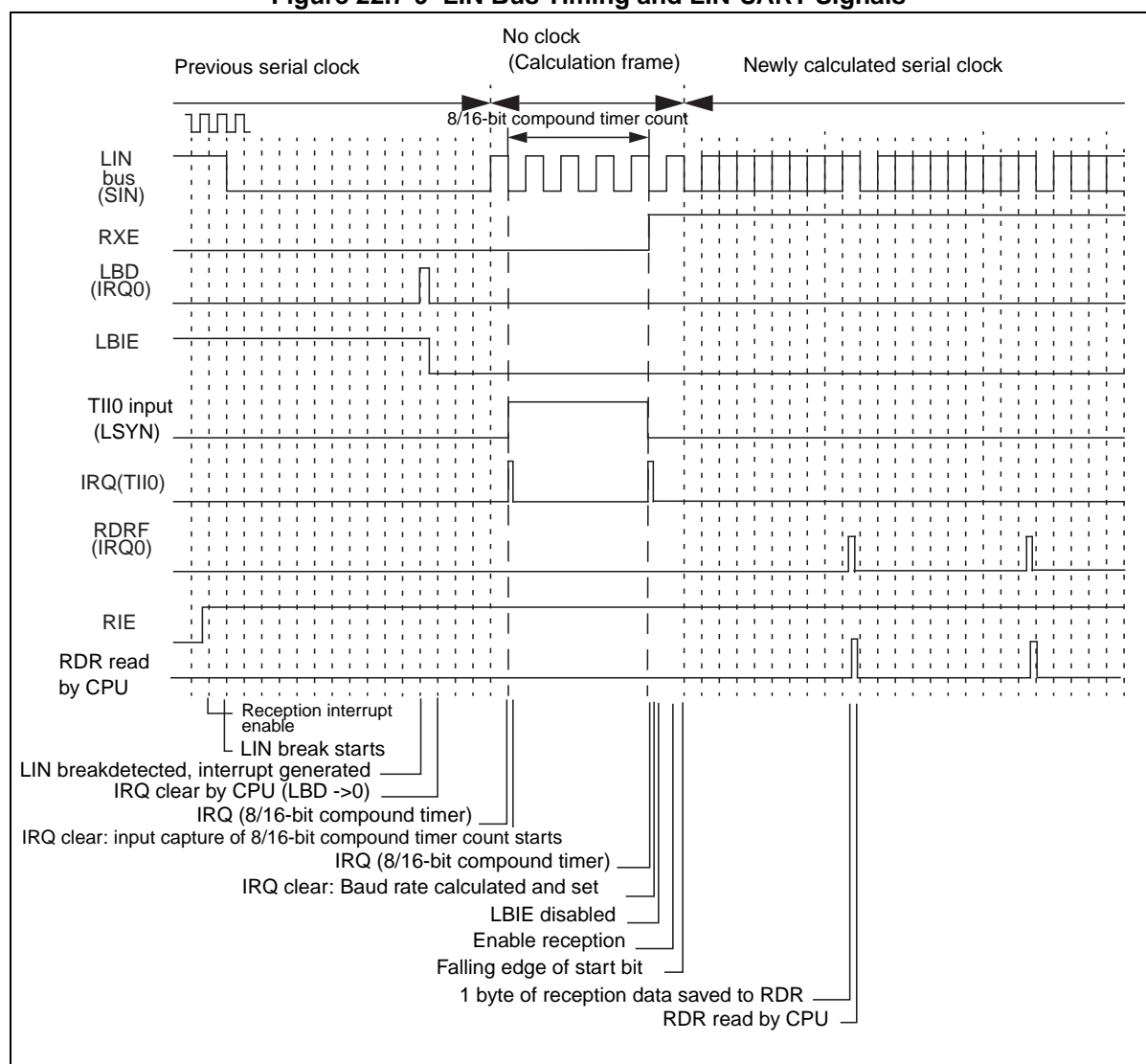


Figure 22.7-8 LIN-UART Operation in LIN Slave Modes



● LIN bus timing

Figure 22.7-9 LIN Bus Timing and LIN-UART Signals



## 22.7.4 Serial Pin Direct Access

---

**Transmission pin (SOT) or reception pin (SIN) can be accessed directly.**

---

### ■ LIN-UART Pin Direct Access

The LIN-UART allows the programmer to directly access the serial I/O pins.

The status of the serial input pin (SIN) can be read by using the serial I/O pin direct access bit (ESCR:SIOP).

You can set the value of the serial output pin (SOT) arbitrarily when the serial output is enabled (SMR:SOE=1) after direct write to the serial output pin (SOT) is enabled (ESCR:SOPE = 1), and then "0" or "1" is written to the serial I/O pin direct access bit (ESCR:SIOP).

In LIN mode, this feature is used for reading transmitted data or for error handling when a LIN bus line signal is physically incorrect.

---

#### Note:

Direct access is allowed only when transmission is not in progress (the transmission shift register is empty).

Before enabling transmission (SMR:SOE = 1), write a value to the serial output pin direct access bit (ESCR:SIOP). This prevents a signal of an unexpected level from being output since the SIOP bit holds a previous value.

While the value of the SIN pin is read by normal read, the value of the SOT pin is read for the SIOP bit by the read-modify-write (RMW) instructions.

---

## 22.7.5 Bi-directional Communication Function (Normal Mode)

Normal serial bi-directional communication can be performed in operation mode 0 or 2. Asynchronous mode and synchronous mode can be selected in operation modes 0 and 2, respectively.

### ■ Bi-directional Communication Function

To operate the LIN-UART in normal mode (operation mode 0 or 2), the settings shown in Figure 22.7-10 are required.

**Figure 22.7-10 Settings of LIN-UART Operation Modes 0 and 2**

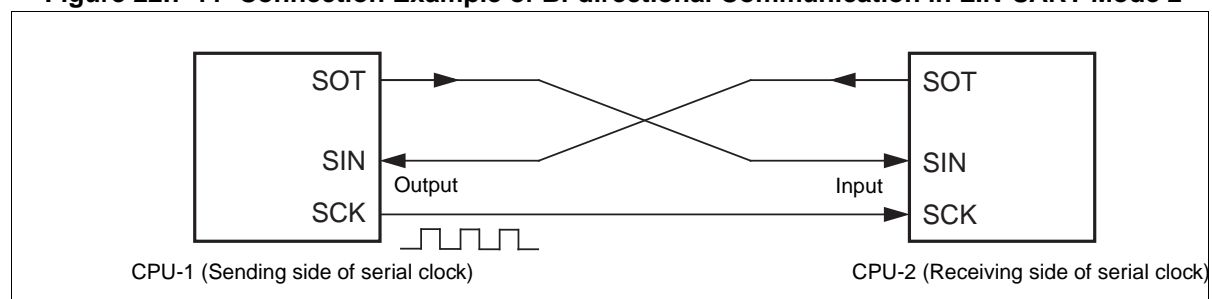
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 0)→	⊙	⊙	⊙	⊙	X	0	⊙	⊙	0	0	0	⊙	0	0	⊙	⊙
(Mode 2)→	☐	☐	☐	+	X	0	⊙	⊙	1	0	⊙	⊙	0	0	⊙	⊙
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set comparison data (during writing) Retain reception data (during reading)							
(Mode 0)→	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙								
(Mode 2)→	☐	⊙	☐	⊙	⊙	⊙	⊙	⊙								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 0)→	X	X	X	X	⊙	⊙	0	0	0	0	X	X	X	0	⊙	⊙
(Mode 2)→	X	X	X	X	⊙	⊙	☐	⊙	0	X	⊙	⊙	⊙	0	☐	☐

⊙ : Used bit  
 X : Unused bit  
 1 : Set to "1"  
 0 : Set to "0"  
 ☐ : Used when SSM = 1 (Synchronous star/stop bit mode)  
 + : Bit correctly set automatically

### ● Inter-CPU connection

For bi-directional communication, interconnect two CPUs as shown in Figure 22.7-11.

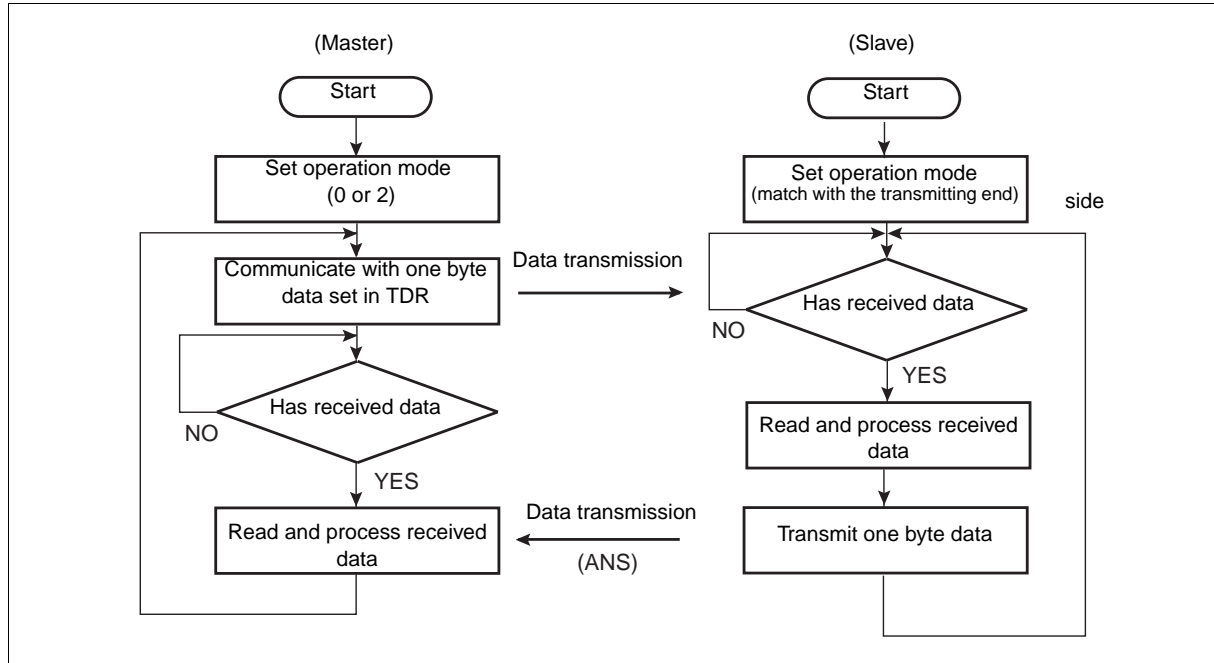
**Figure 22.7-11 Connection Example of Bi-directional Communication in LIN-UART Mode 2**



● Communication procedure example

The communication is started from transmitting end at arbitrary timing when data is ready to be transmitted. When transmission data is received in the receiving side, ANS (1 byte in the example) is returned on a regular basis. Figure 22.7-12 shows an example of bi-directional communication flowchart.

**Figure 22.7-12 Example of Bi-directional Communication Flowchart**



## 22.7.6 Master/slave Mode Communication Function (Multi-processor Mode)

Operation mode 1 allows communication between multiple CPUs connected in master/slave mode. It can be used as a master or slave.

### ■ Master/Slave Mode Communication Function

To operate the LIN-UART in multiprocessor mode (operation mode 1), the settings shown in Figure 22.7-13 are required.

**Figure 22.7-13 Settings of LIN-UART Operation Mode 1**

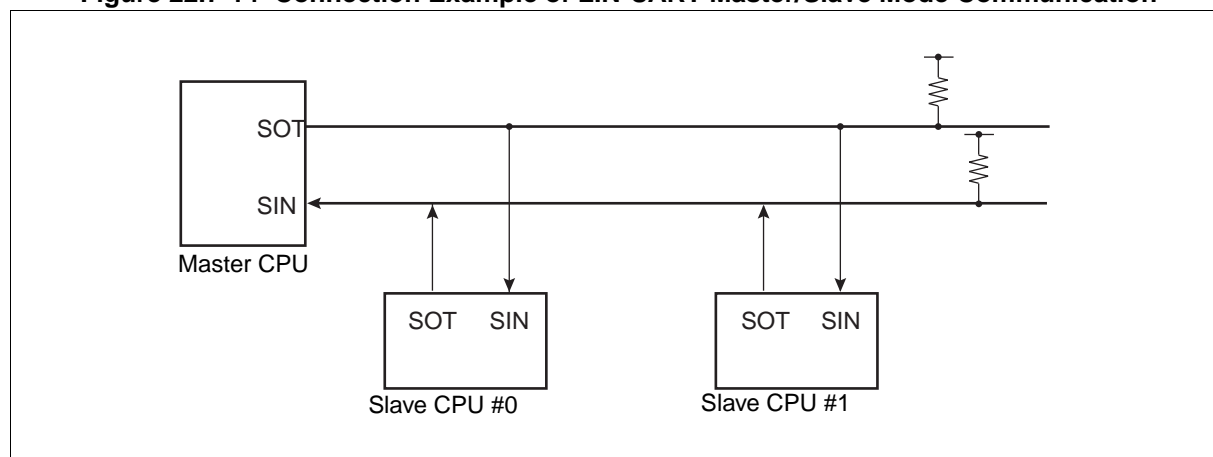
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 1)→	+	X	⊙	⊙	⊙	0	⊙	⊙	0	1	0	⊙	0	0	⊙	⊙
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)							
(Mode 1)→	X	⊙	⊙	⊙	⊙	⊙	⊙	⊙								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 1)→	X	X	X	X	⊙	⊙	0	0	0	X	X	X	X	0	⊙	⊙

⊙ : Used bit  
 X : Unused bit  
 1 : Set to "1"  
 0 : Set to "0"  
 + : Bit correctly set automatically

#### ● Inter-CPU Connection

For master/slave mode communication, a communication system is configured by connecting between one master CPU and multiple slave CPUs with two common communication lines, as shown in Figure 22.7-14. The LIN-UART can be used as the master or slave.

**Figure 22.7-14 Connection Example of LIN-UART Master/Slave Mode Communication**



### ● Function Selection

For master/slave mode communication, select the operation mode and the data transfer method, as shown in Table 22.7-4.

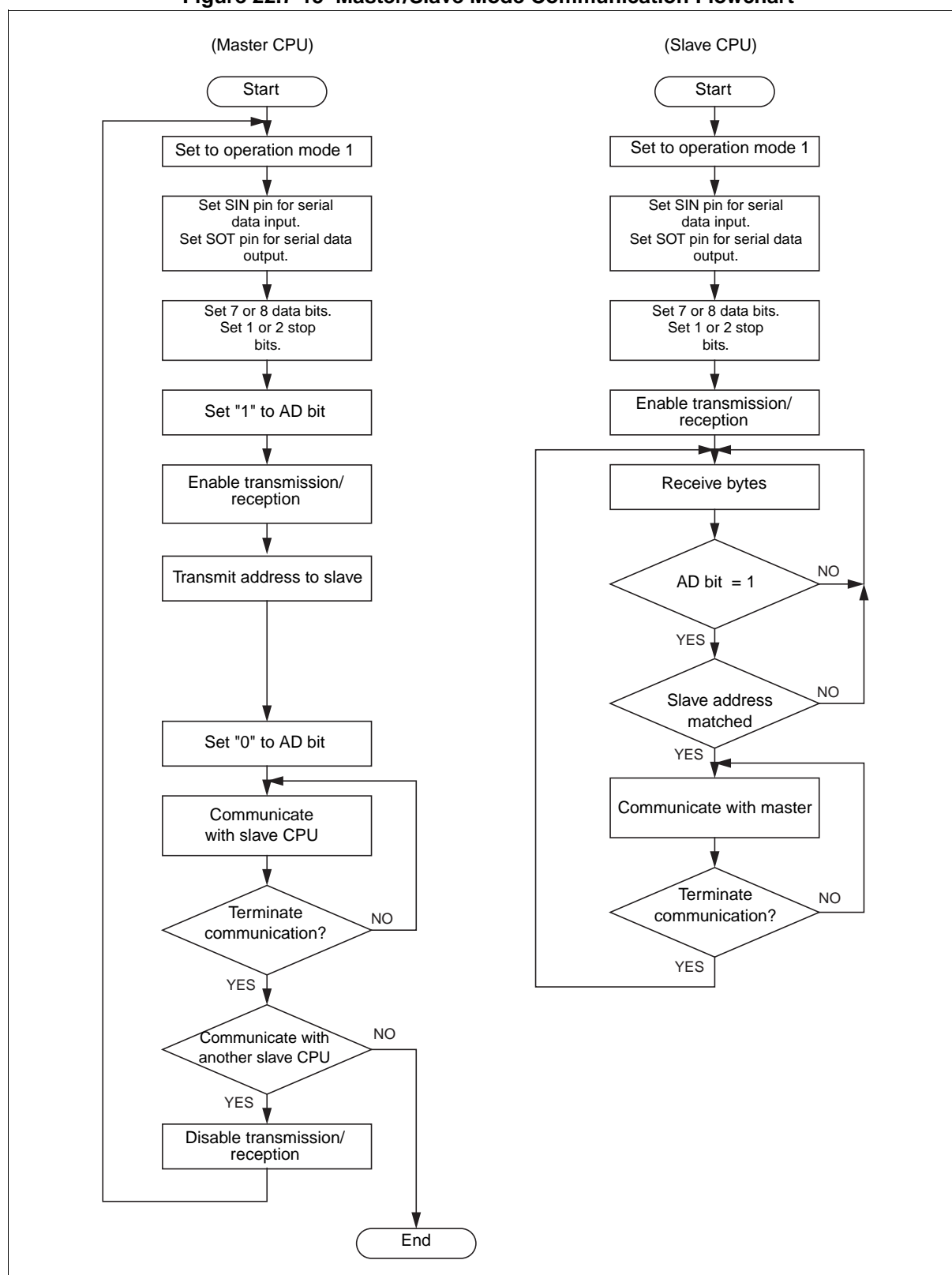
**Table 22.7-4 Select of Master/Slave Mode Communication Function**

	Operation mode		Data	Parity	Synchronous method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission/reception	Mode 1 (AD bit transmission and reception)	Mode 1 (AD bit transmission and reception)	AD = 1 + 7-bit or 8-bit address	None	Asynchronous	1 bit or 2 bits	LSB first or MSB first
Data transmission/reception			AD = 0 + 7-bit or 8-bit data				

### ● Communication procedure

Communication is started by transmitting address data from the master CPU. The address data, whose AD bit is set as "1", determines the slave CPU to be the destination. Each slave CPU checks address data by using a program, and communicates with the master CPU when the data matches an assigned address.

Figure 22.7-15 shows a flowchart for master/slave mode communication.

**Figure 22.7-15 Master/Slave Mode Communication Flowchart**

## 22.7.7 LIN Communication Function

For LIN-UART communication, a LIN device can be used in the LIN master system or the LIN slave system.

### ■ LIN Master/Slave Mode Communication Function

Figure 22.7-16 shows the required settings for the LIN communication mode of LIN-UART (operation mode 3).

**Figure 22.7-16 Settings of LIN-UART Operation Mode 3 (LIN)**

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR, SMR	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
(Mode 3)→	+	X	+	+	X	0	⊙	⊙	1	1	0	⊙	0	0	⊙	⊙
SSR, RDR/TDR	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set conversion data (during writing) Retain reception data (during reading)							
(Mode 3)→	X	⊙	⊙	⊙	⊙	+	⊙	⊙								
ESCR, ECCR	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SCES	Reserved	LBR	MS	SCDE	SSM	Reserved	RBI	TBI
(Mode 3)→	⊙	⊙	⊙	⊙	⊙	⊙	0	0	0	⊙	X	X	X	0	⊙	⊙

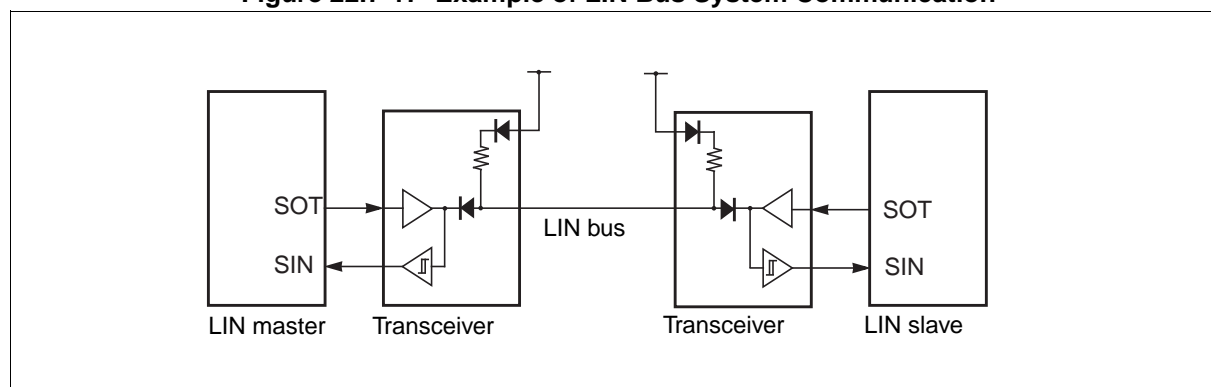
⊙ : Used bit  
 X : Unused bit  
 1 : Set to "1"  
 0 : Set to "0"  
 + : Bit correctly set automatically

#### ● LIN device connection

Figure 22.7-17 shows an example of the LIN bus system communication.

The LIN-UART can serve as the LIN master or LIN slave.

**Figure 22.7-17 Example of LIN Bus System Communication**



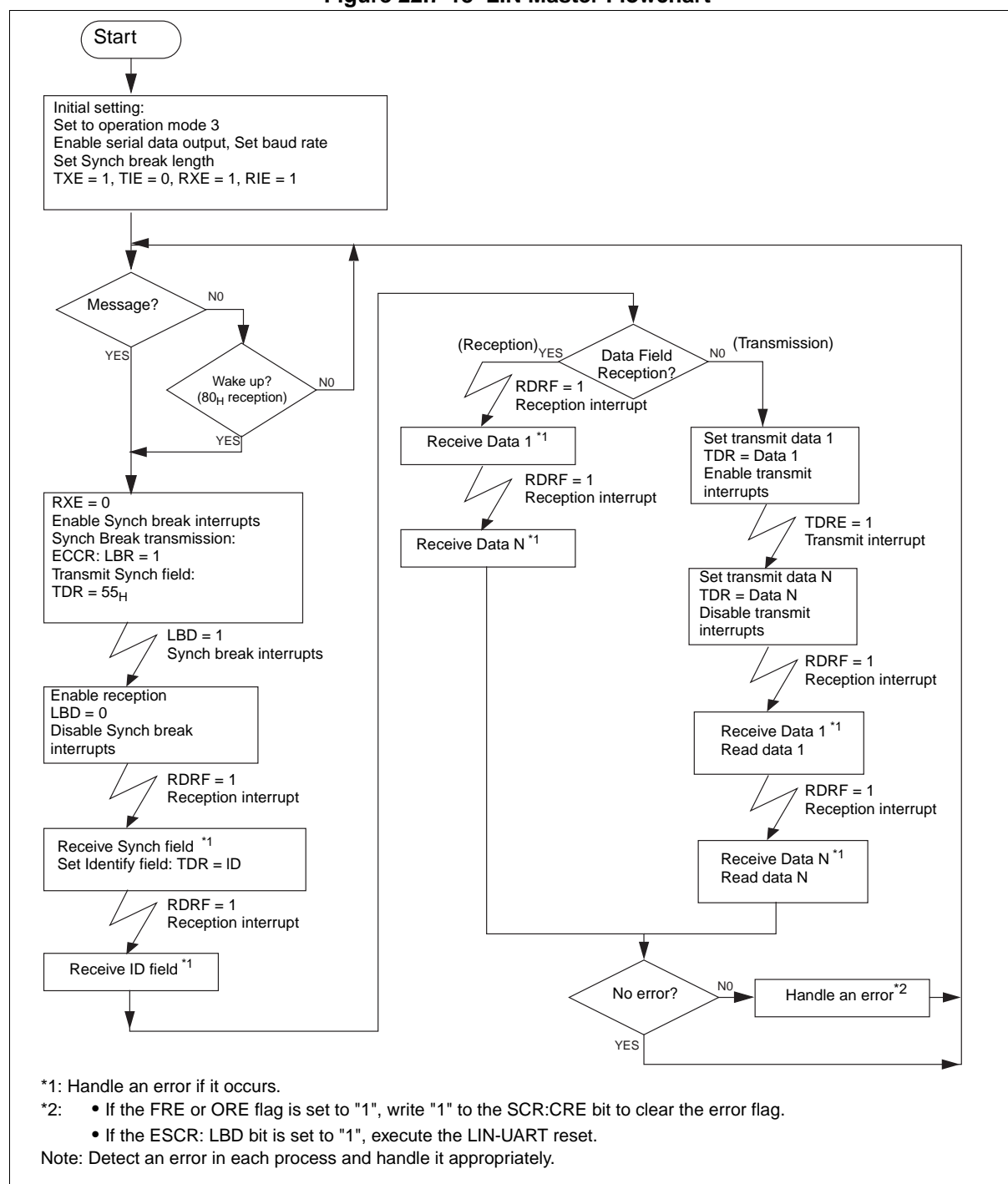


## 22.7.8 Example of LIN-UART LIN Communication Flowchart(Operation Mode 3)

This section shows examples of LIN-UART LIN communication flowchart.

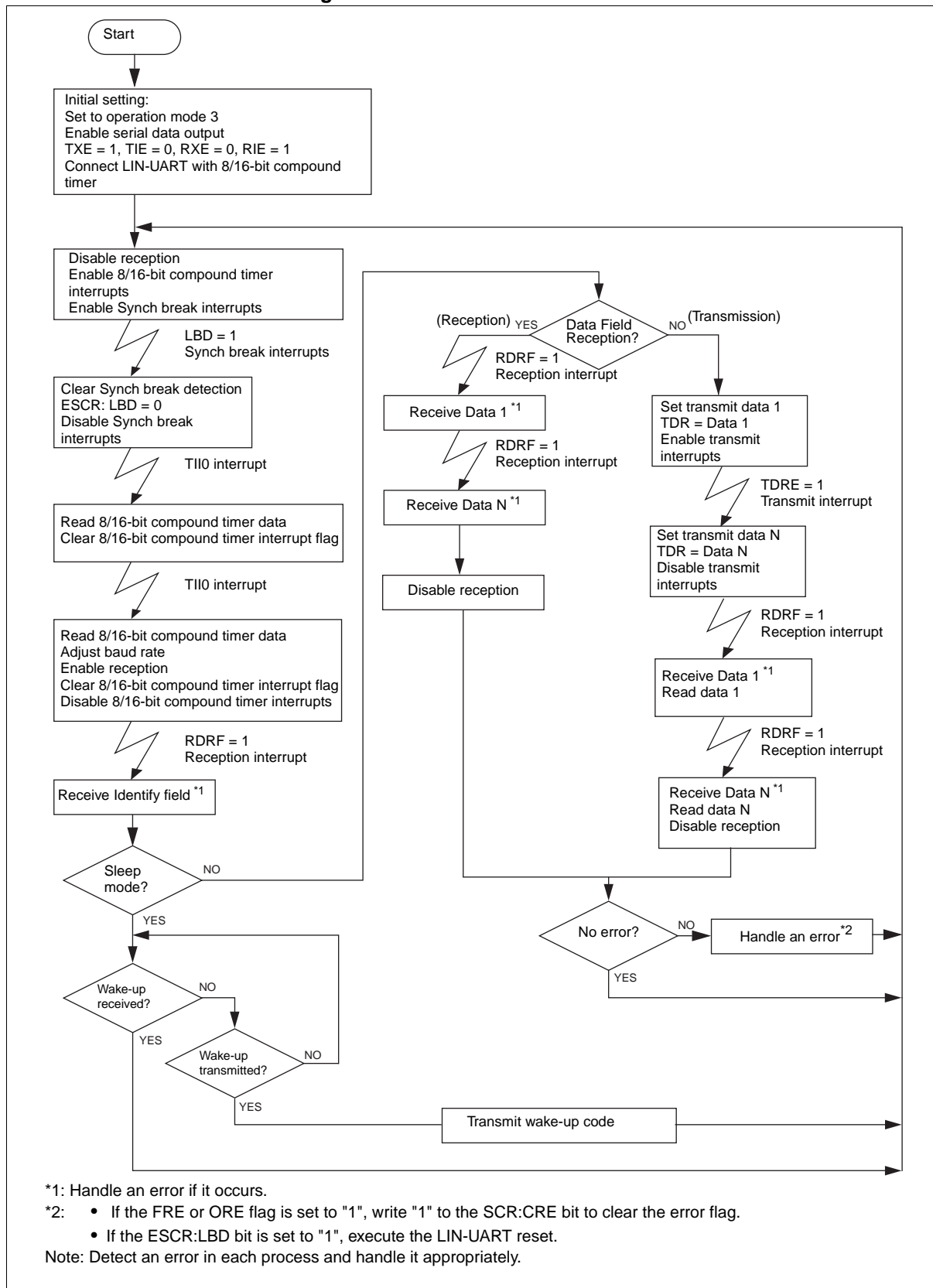
### ■ LIN Master Device

Figure 22.7-18 LIN Master Flowchart



## ■ LIN Slave Device

Figure 22.7-19 LIN Slave Flowchart



## 22.8 Notes on Using LIN-UART

---

**This section shows notes on using the LIN-UART.**

---

### ■ Notes on Using LIN-UART

#### ● Enabling operation

The LIN-UART has the TXE (transmission) and RXE (reception) enable bit in the LIN-UART serial control register (SCR) for transmission and reception, respectively. Since both transmission and reception are disabled by default (internal value), these operations must be enabled before transfer. Also, you can disable these operations to stop transfer as required.

#### ● Setting communication mode

The communication mode must be set while the LIN-UART is stopped. If the mode is set during transmission/reception, the transmitted/received data is not guaranteed.

#### ● Timing of enabling transmit interrupts

Since the default (initial) value of the transmit data empty flag bit (SSR:TDRE) is "1" (no transmit data, transmit data write enabled), a transmit interrupt request is generated immediately when transmit interrupt request is enabled (SSR:TIE =1). To prevent this, be sure to set the transmit data before setting the TIE flag to "1".

#### ● Changing operation setting

Reset the LIN-UART after changing its settings, such as adding the start/stop bit or changing the data format.

The correct operation settings are not guaranteed even if you reset the LIN-UART (SMR:UPCL = 1) concurrently with setting the LIN-UART serial mode register (SMR). Therefore, after setting the bit in LIN-UART serial mode register (SMR), reset the LIN-UART (SMR:UPCL = 1) again.

#### ● Using LIN function

Although the LIN functions are available in the mode 3, the LIN format is automatically set in the mode 3 (8-bit data, no parity, 1 stop bit, LSB-first).

While the length of LIN break transmit bit is variable, the detection bit length is fixed to 11 bits.

#### ● Setting LIN slave

When starting LIN slave mode, be sure to set the baud rate before receiving the LIN synch break in order to make sure that the minimum 13 bits length of the LIN synch break is detected.

#### ● Bus idle function

The bus idle is not available in synchronous mode 2.

● AD bit (LIN-UART serial control register (SCR): Address/data format selection bit)

Be sure to note the followings when using the AD bit.

The AD bit is used to select the address/data for transmission when it is written, and to read the AD bit received last when it reads. Internally, the AD bit values for transmission and reception are stored in separate registers.

The transmit AD bit value is read when read-modify-write (RMW) instructions are used. Therefore, an incorrect value may be written to the AD bit when another bit in the SCR is bit-accessed.

For the above reason, the AD bit must be set at the last access to the SCR before transmission. Or, the above problem can be prevented by byte-accessing whenever the SCR is written.

● LIN-UART software reset

Execute the LIN-UART software reset (SMR:UPCL = 1) when the TXE bit in the LIN-UART serial control register (SCR) is "0".

● Synch break detection

In mode 3 (LIN mode), when serial input has 11 bits width or more and becomes "L", the LBD bit in the extended status control register (ESCR) is set to "1" (Synch break detection) and the LIN-UART waits for the Synch field. As a result, when serial input has more than 11 bits of "0" except Synch break, the LIN-UART recognizes that the Synch break is input (LBD = 1), and then waits for the Synch field.

In this case, execute the LIN-UART reset (SMR:UPCL = 1).

● Handling framing errors

1) (CRE resets reception state machine and next falling edge at SINn starts reception of new byte (Figure 22.8-1).

In order to avoid desynchronization of the data stream, it is necessary to set the CRE bit within a half-bit time immediately after an error is received (as shown in Figure 22.8-2), or to wait for the application-dependent time while SINn is idling after an error is received.

2) If a framing error occurs (stop bit: SINn= "0") and the next start bit (SINn= "0") immediately follows it, this start bit is recognized regardless of a falling edge for the start bit and reception is started. This sequence is used for detecting the continuous "L" state of the serial data input (SINn) when the next framing error is detected while the data stream is synchronized (See "When reception is always enabled (RXE=1)" in Figure 22.8-3).

If this operation is not necessary, disable data reception temporarily after receiving a framing error (RXE = 1 → 0 → 1). Therefore, the falling edge of the serial data input (SINn) is detected, the start bit is recognized when "L" is detected at the reception sampling point, and the reception is started (See "When reception is temporarily disabled (RXE=1→0→1)" in Figure 22.8-3).

**Figure 22.8-1 CRE bit timing**

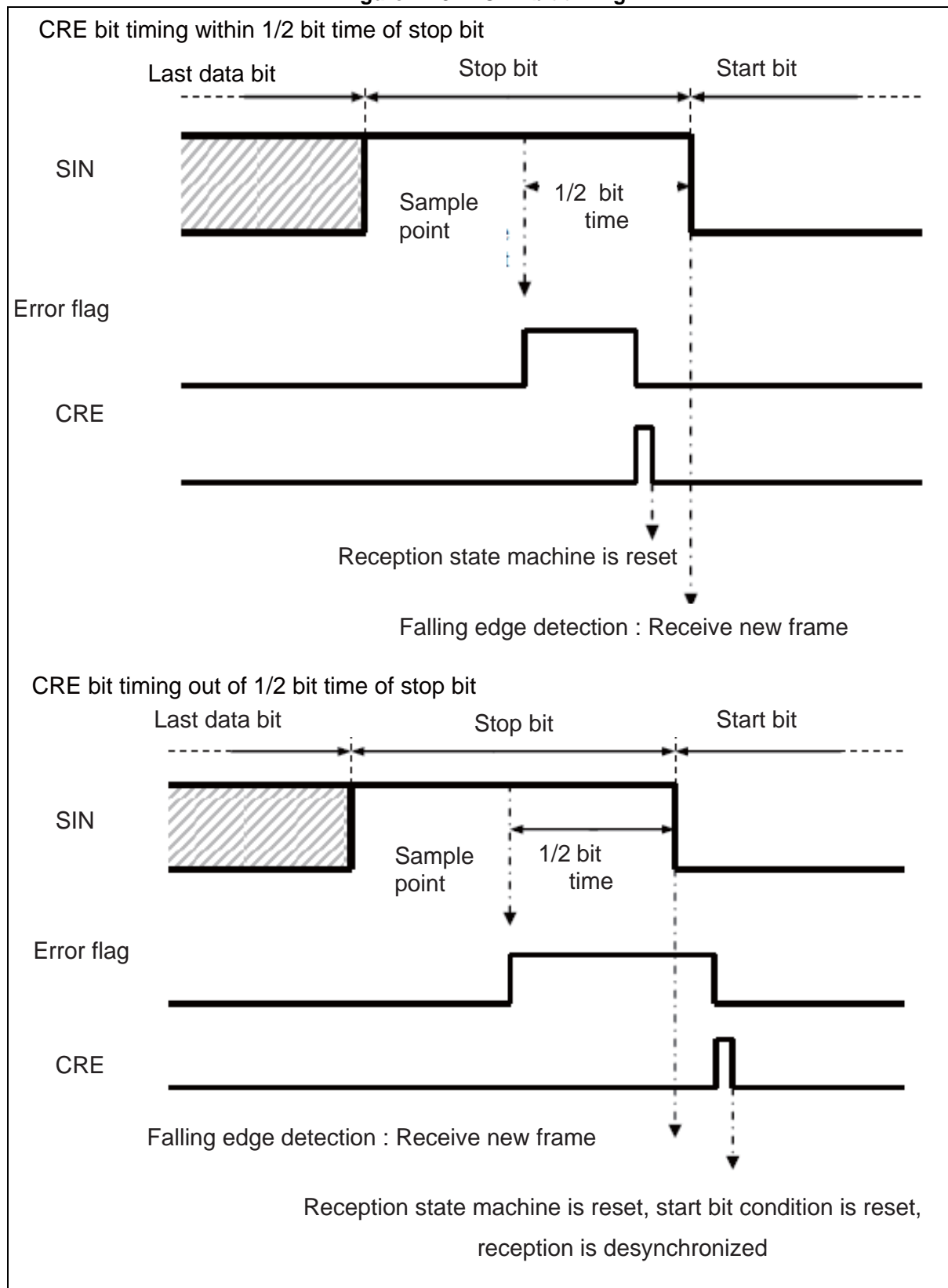
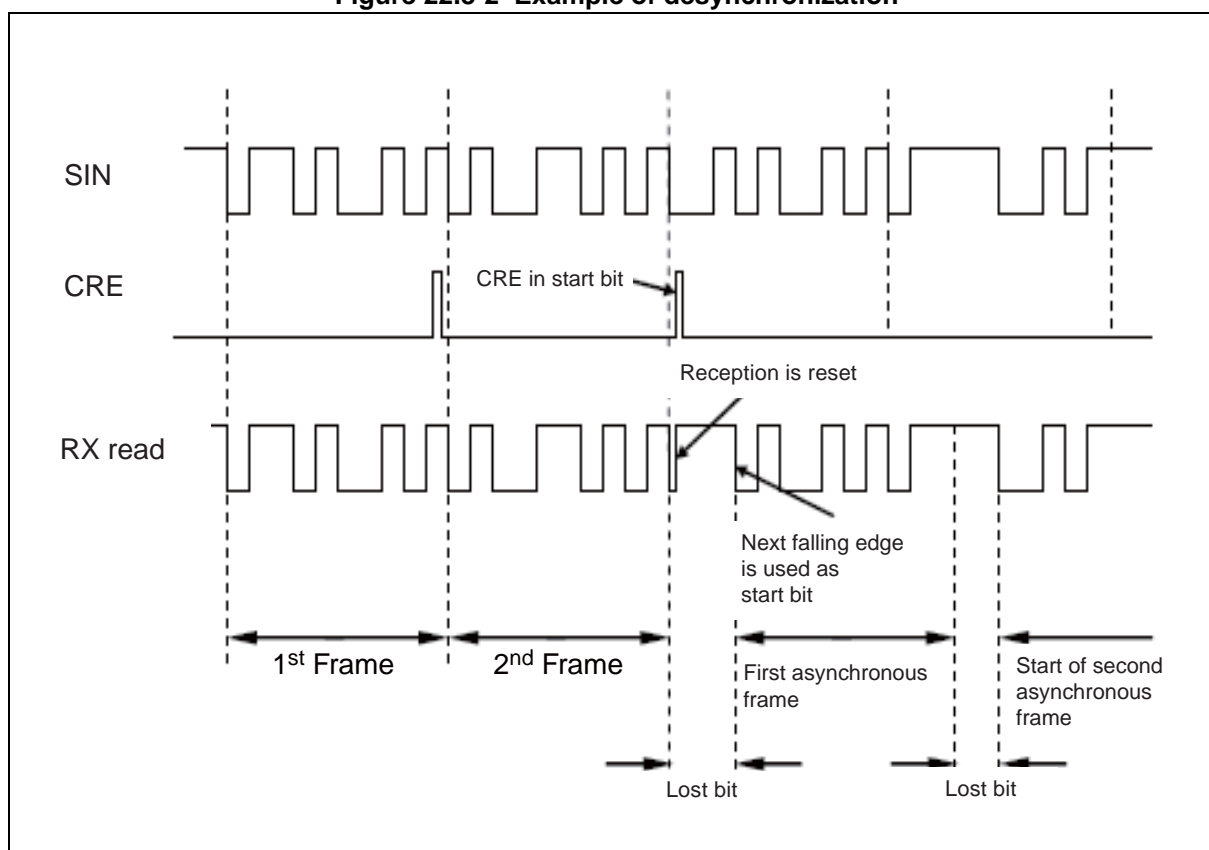
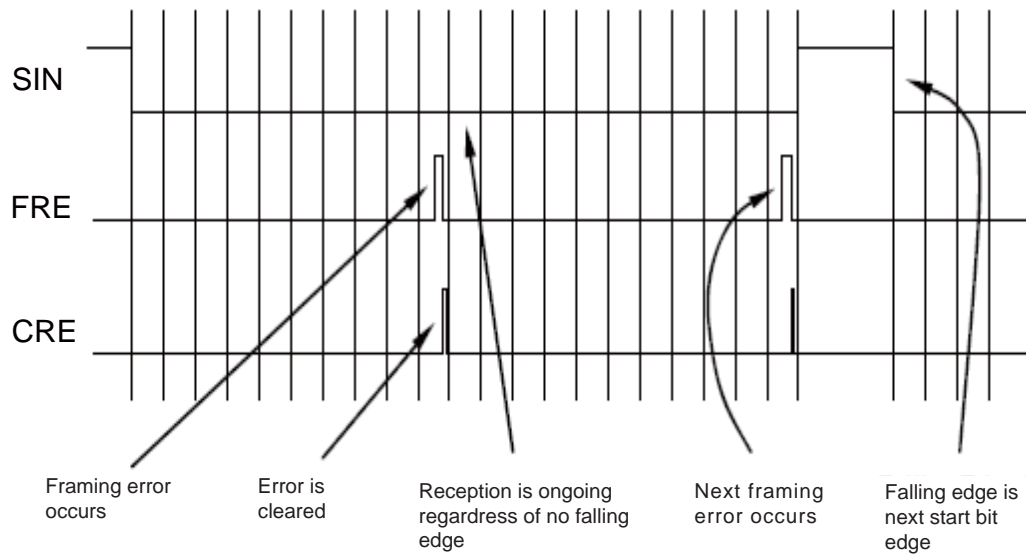


Figure 22.8-2 Example of desynchronization

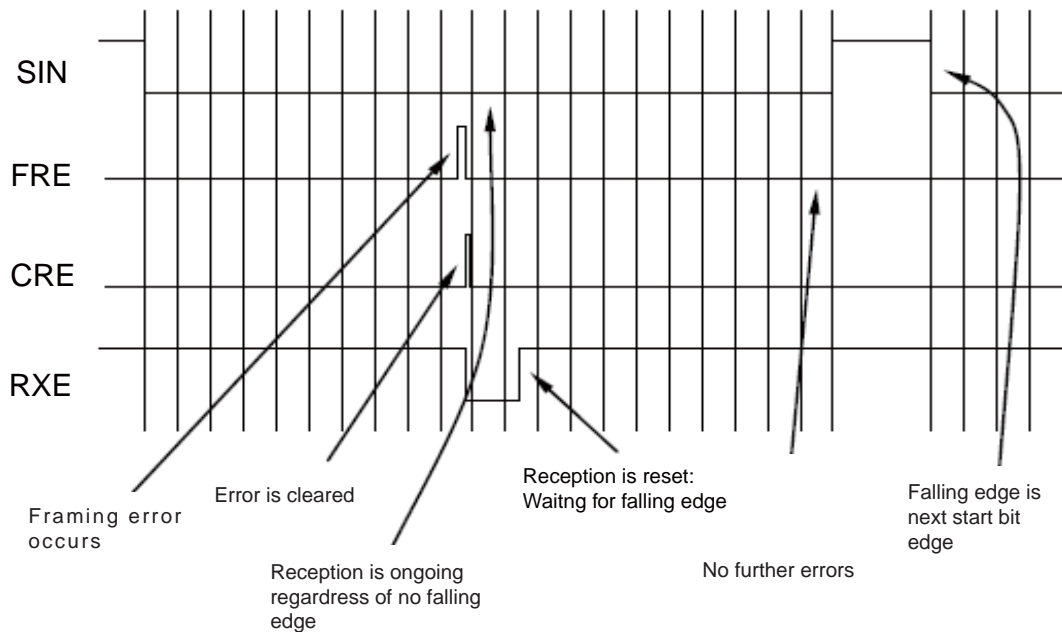


**Figure 22.8-3 UART dominant bus operation**

When reception is always enabled (RXE=1)



When reception is temporarily disabled (RXE=1 → 0 → 1)



## 22.9 Sample Programs of LIN-UART

This section provides sample programs for operating the LIN-UART.

### ■ Sample Programs of LIN-UART

For sample programs of LIN-UART, see "■Sample Programs" in Preface.

### ■ Setting Methods not Covered by Sample Programs

#### ● How to select the operation mode

Use the operation mode selection (SMR.MD[1:0]).

Operation mode		Operation mode selection (MD[1:0]).
Mode 0	Normal (Asynchronous)	Set to "00 <sub>B</sub> "
Mode 1	Multiprocessor	Set to "01 <sub>B</sub> "
Mode 2	Normal (Synchronous)	Set to "10 <sub>B</sub> "
Mode 3	LIN	Set to "11 <sub>B</sub> "

#### ● Operation clock types and how to select it

Use the external clock select bit (SMR.EXT).

Clock input	External clock select bit (EXT)
To select a dedicated baud rate generator	Set to "0"
To select an external clock	Set to "1"

#### ● How to control the SCK, SIN, and SOT pins

Use the following setting.

	LIN-UART
To set the SCK pin as input	DDR6.P65 =0 SMR.SCKE =0
To set the SCK pin as output	SMR.SCKE =1
To use the SIN pin	DDR6.P67 =0
To use the SOT pin	SMR.SOE =1



● How to enable/disable the LIN-UART operation

Use the reception enable bit (SCR.RXE).

Control item	Reception enable bit (RXE)
Disable reception	Set to "0"
Enable reception	Set to "1"

Use the transmit control bit (SCR.TXE).

Control item	Transmit control bit (TXE)
Disable transmission	Set to "0"
Enable transmission	Set to "1"

● How to use an external clock as the LIN-UART serial clock

Use the one-to-one external clock enable bit (SMR.OTO).

Control item	Reception enable bit (OTO)
Enable external clock	Set to "1"

● How to restart the reload counter

Use the reload counter restart bit (SMR.REST).

Control item	Reload counter restart bit (REST)
Restart the reload counter	Set to "1"

● How to reset the LIN-UART

Use the LIN-UART programmable clear bit (SMR.UPCL).

Control item	LIN-UART programmable clear bit (UPCL)
Reset the LIN-UART software	Set to "1"

● How to set the parity

Use the parity enable bit (SCR.PEN) and the parity select bit (SCR.P).

Operation	Parity control (PEN)	Parity polarity (P)
To set to no parity	Set to "0"	—
To set to even parity	Set to "1"	Set to "0"
To set to odd parity	Set to "1"	Set to "1"

● How to set the data length

Use the data length select bit (SCR.CL).

Operation	Data length select bit (CL)
To set the bit length to 7	Set to "0"
To set the bit length to 8	Set to "1"

● How to select the STOP bit length

Use the STOP bit length control (SCR.SBL).

Operation	STOP bit length control (SBL)
To set STOP bit length to 1	Set to "0"
To set STOP bit length to 2	Set to "1"

● How to clear the error flag

Use the reception error flag clear bit (SCR.CRE).

Control item	Reception error flag clear bit (CRE)
To clear the error flag (PE, ORE, FRE)	Set to "0"

● How to set the transfer direction

Use the transfer direction selection bit (SSR.BDS).

LSB/MSB can be selected for transfer direction in any operation mode.

Control item	Serial data direction control (BDS)
To select the LSB first transfer (from the least significant bit)	Set to "0"
To select the MSB first transfer (from the most significant bit)	Set to "1"

● How to clear the reception completion flag

Uses the following setting.

Control item	Method
To clear the reception completion flag	Read the RDR register

The first RDR register read is the reception initiation.

● How to clear the transmit buffer empty flag

Uses the following setting.

Control item	Method
To clear the transmit buffer empty flag	Write to TDR register

The first TDR register write is the transmit initiation.

● How to select the data format (Address/Data) (Only in mode 1)

Use the address/data selection bit (SCR:AD).

Operation	Address/Data select bit (AD)
To select the data frame	Set to "0"
To select the address frame	Set to "1"

This is effective only at transmission. The AD bit is ignored at reception.

● How to set the baud rate

See Section "22.6 LIN-UART Baud Rate".

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

	Interrupt level setting register	Interrupt vector
Reception	Interrupt level register (ILR1) Address: 0007A <sub>H</sub>	#7 Address: 0FFFC <sub>H</sub>
Transmission	Interrupt level register (ILR2) Address: 0007B <sub>H</sub>	#8 Address: 0FFEA <sub>H</sub>

● How to enable/disable/clear interrupts

Interrupt request enable bit (SSR.RIE), (SSR.TIE) is used to enable interrupts.

Operation	UART reception	UART transmission
	Reception interrupt enable bit (RIE)	Reception interrupt enable bit (TIE)
To disable interrupt requests	Set to "0"	
To enable interrupt requests	Set to "1"	

The following setting is used to clear interrupt requests.

Operation	UART reception	UART transmission
To clear interrupt requests	The reception data register full (RDRF) is cleared by reading the LIN-UART serial input register (RDR).	The transmit data register empty (TDRE) is set to "0" by writing data to the serial output data register (TDR).
	The error flags (PE, ORE, FRE) are set to "0" by writing "1" to the error flag clear bit (CRE).	



# CHAPTER 23

---

## I<sup>2</sup>C

**This chapter describes functions and operations of the I<sup>2</sup>C.**

- 23.1 Overview of I<sup>2</sup>C
- 23.2 I<sup>2</sup>C Configuration
- 23.3 I<sup>2</sup>C Channels
- 23.4 I<sup>2</sup>C Bus Interface Pins
- 23.5 I<sup>2</sup>C Registers
- 23.6 I<sup>2</sup>C Interrupts
- 23.7 I<sup>2</sup>C Operations and Setup Procedure Examples
- 23.8 Notes on Use of I<sup>2</sup>C
- 23.9 Sample Programs for I<sup>2</sup>C

## 23.1 Overview of I<sup>2</sup>C

The I<sup>2</sup>C interface supports the I<sup>2</sup>C bus specification published by Philips. The interface provides the functions of transmission and reception in master and slave modes, detection of arbitration lost, detection of slave address and general call address, generation and detection of start and stop conditions, bus error detection, and MCU standby wakeup.

### ■ I<sup>2</sup>C Functions

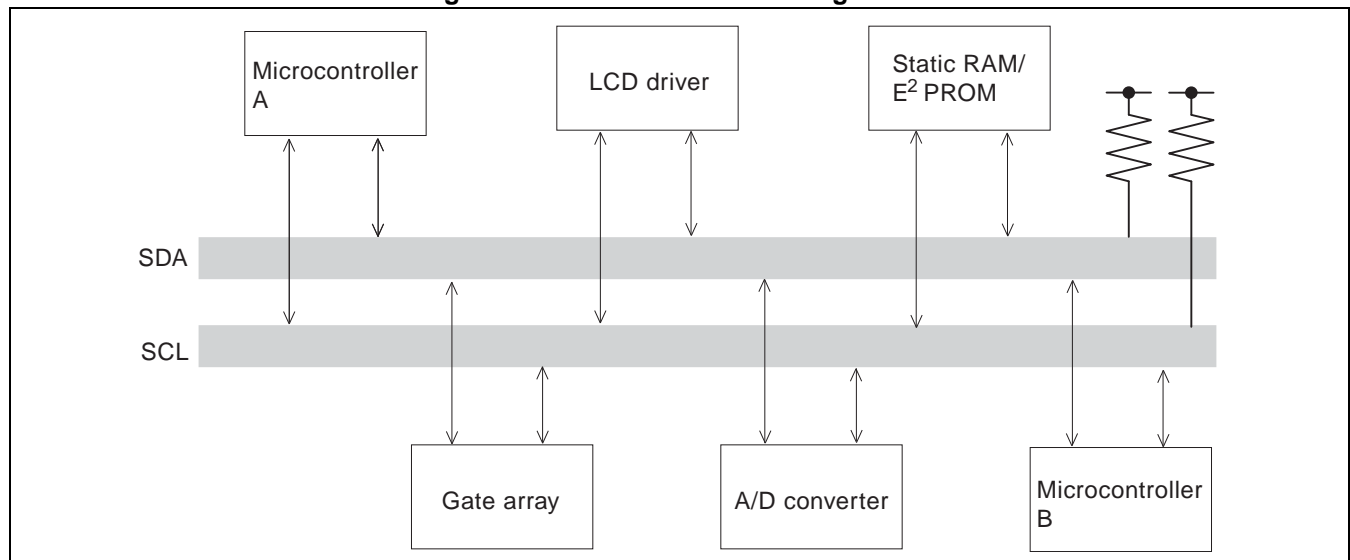
The I<sup>2</sup>C interface is a two-wire, bi-directional bus consisting of a serial data line (SDA) and serial clock line (SCL). The devices connected to the bus via these two wires can exchange data, and each device can operate as a sender or receiver in accordance with their respective functions based on the unique address assigned to each device. Furthermore, the interface establishes a master/slave relationship between devices.

Also, the I<sup>2</sup>C interface can connect multiple devices provided the bus capacitance does not exceed an upper limit of 400 pF. The I<sup>2</sup>C interface is a true multi-master bus with collision detection and a communication control protocol that prevent loss of data even if more than one master attempts to start a data transfer at the same time.

The communication control protocol ensures that only one master is able to take control of the bus at a time, even if multiple masters attempt to take control of the bus simultaneously, without messages being lost or data being altered. Multi-master means that more than one master can attempt to take control of the bus at the same time without causing messages to be lost.

Also, the I<sup>2</sup>C interface includes a function to wake up the MCU from standby mode.

Figure 23.1-1 I<sup>2</sup>C Interface Configuration



## 23.2 I<sup>2</sup>C Configuration

---

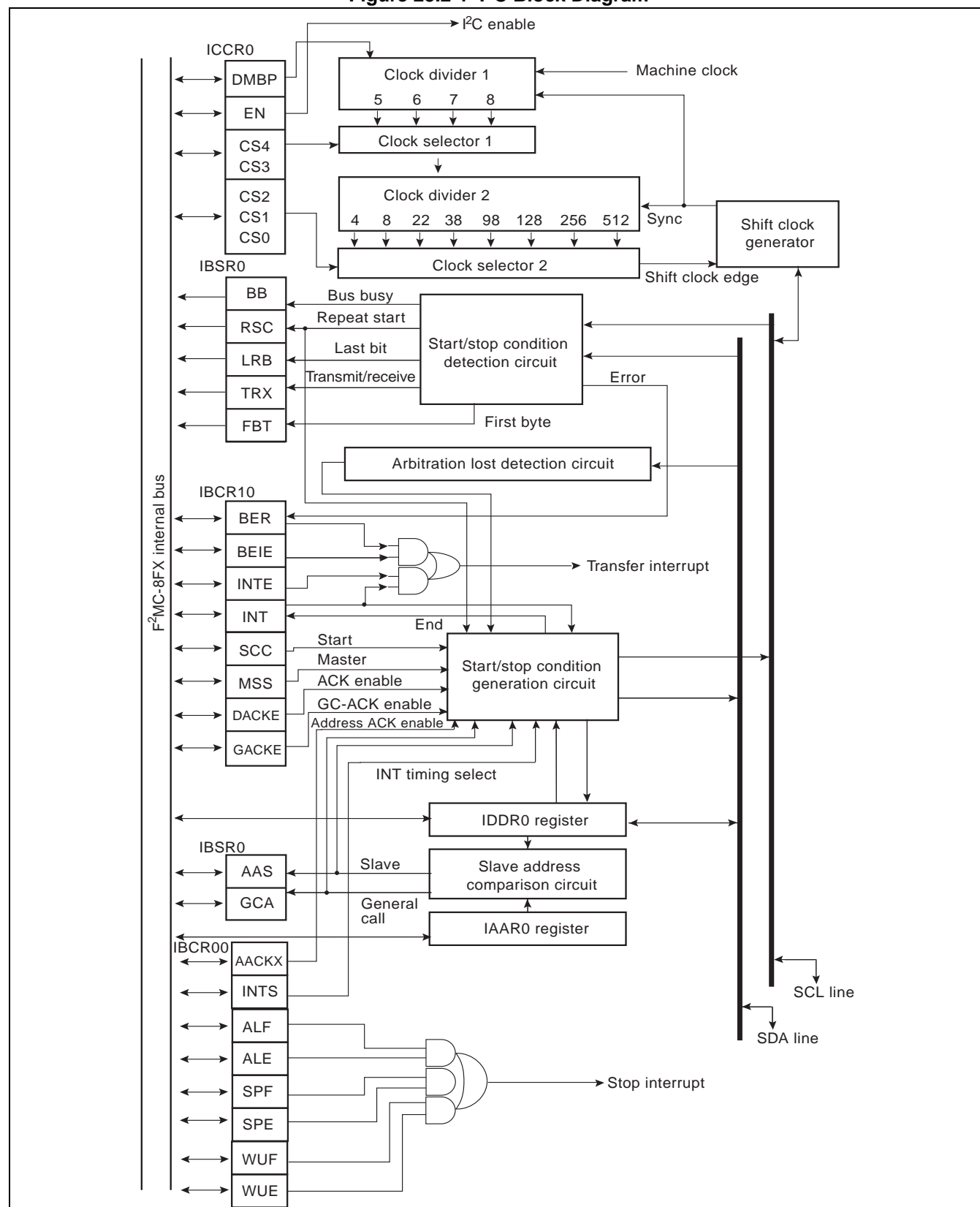
I<sup>2</sup>C consists of the following blocks:

- Clock selector
  - Clock divider
  - Shift clock generator
  - Start/stop condition generation circuit
  - Start/stop condition detection circuit
  - Arbitration lost detection circuit
  - Slave address comparison circuit
  - IBSR register
  - IBCR registers (IBCR00, IBCR10)
  - ICCR0 register
  - IAAR0 register
  - IDDR0 register
-



## I<sup>2</sup>C Block Diagram

Figure 23.2-1 I<sup>2</sup>C Block Diagram



- Clock selector, clock divider, and shift clock generator

This circuit uses the machine clock to generate the shift clock for the I<sup>2</sup>C bus.

- Start/stop condition generation circuit

When a start condition is transmitted with the bus idle (SCL0 and SDA0 at the "H" level), a master starts communications. When SCL0 = "H", a start condition is generated by changing the SDA0 line from "H" to "L". The master can terminate its communication by generating a stop condition. When SCL0 = "H", a stop condition is generated by changing the SDA0 line from "L" to "H".

- Start/stop condition detection circuit

This circuit detects a start/stop condition for data transfer.

- Arbitration lost detection circuit

This interface circuit supports multi-master systems. If two or more masters attempt to transmit at the same time, the arbitration lost condition (if logic level "1" is sent when the SDA0 line goes to the "L" level) occurs. When the arbitration lost is detected, IBCR00:ALF is set to "1" and the master changes to a slave automatically.

- Slave address comparison circuit

The slave address comparison circuit receives the slave address after the start condition to compare it with its own slave address. The address is seven-bit data followed by a data direction (R/W) bit in the eighth bit position. If the received address matches the own slave address, the comparison circuit transmits an acknowledgment.

- IBSR0 register

The IBSR0 register shows the status of the I<sup>2</sup>C interface.

- IBCR registers (IBCR00, IBCR10)

The IBCR registers are used to select the operating mode and to enable or disable interrupts, acknowledgment, general call acknowledgment, and the function to wake up the MCU from standby mode.

- ICCR0 register

The ICCR0 register is used to enable I<sup>2</sup>C interface operations and select the shift clock frequency.

- IAAR0 register

The IAAR0 register is used to set the slave address.

- IDDR0 register

The IDDR0 register holds the transmit or receive shift data or address. When transmitted, the data or address written to this register is transferred from the MSB first to the bus.

## ■ Input Clock

I<sup>2</sup>C uses the machine clock as the input clock (shift clock).

## 23.3 I<sup>2</sup>C Channels

---

This section describes the I<sup>2</sup>C channels.

---

### ■ I<sup>2</sup>C Channels

MB95110B/M series contains 1 channel of I<sup>2</sup>C.

Table 23.3-1 and Table 23.3-2 show the correspondence among the channels, pins, and registers respectively.

**Table 23.3-1 I<sup>2</sup>C Pins**

Channel	Pin name	Pin function
0	SCL0 SDA0	I <sup>2</sup> C bus I/O

**Table 23.3-2 I<sup>2</sup>C Registers**

Channel	Register name	Register designation (Representation in this manual)
0	IBCR00	I <sup>2</sup> C bus control register 0
	IBCR10	I <sup>2</sup> C bus control register 1
	IBSR0	I <sup>2</sup> C bus status register
	IDDR0	I <sup>2</sup> C data register
	IAAR0	I <sup>2</sup> C address register
	ICCR0	I <sup>2</sup> C clock control register

## 23.4 I<sup>2</sup>C Bus Interface Pins

---

**This section describes the pins of the I<sup>2</sup>C bus interface and gives their block diagram.**

---

### ■ Pins Related to I<sup>2</sup>C Bus Interface

The pins related to the I<sup>2</sup>C bus interface are the SDA0 and SCL0 pins.

#### ● SDA0 pin

The SDA0 pin can serve as a general-purpose I/O port, external interrupt input (hysteresis input), serial data output pin (N-ch open-drain) for 8-bit serial I/O, and I<sup>2</sup>C data I/O pin (SDA0).

SDA0: When I<sup>2</sup>C is enabled (ICCR0:EN = 1), the SDA0 pin is automatically set as a data I/O pin to function as the SDA0 pin.

To use it as an input pin, enable the I<sup>2</sup>C operation (ICCR0: EN = 1) and write "0" to the corresponding of bit4 port direction register (DDR).

#### ● SCL0 pin

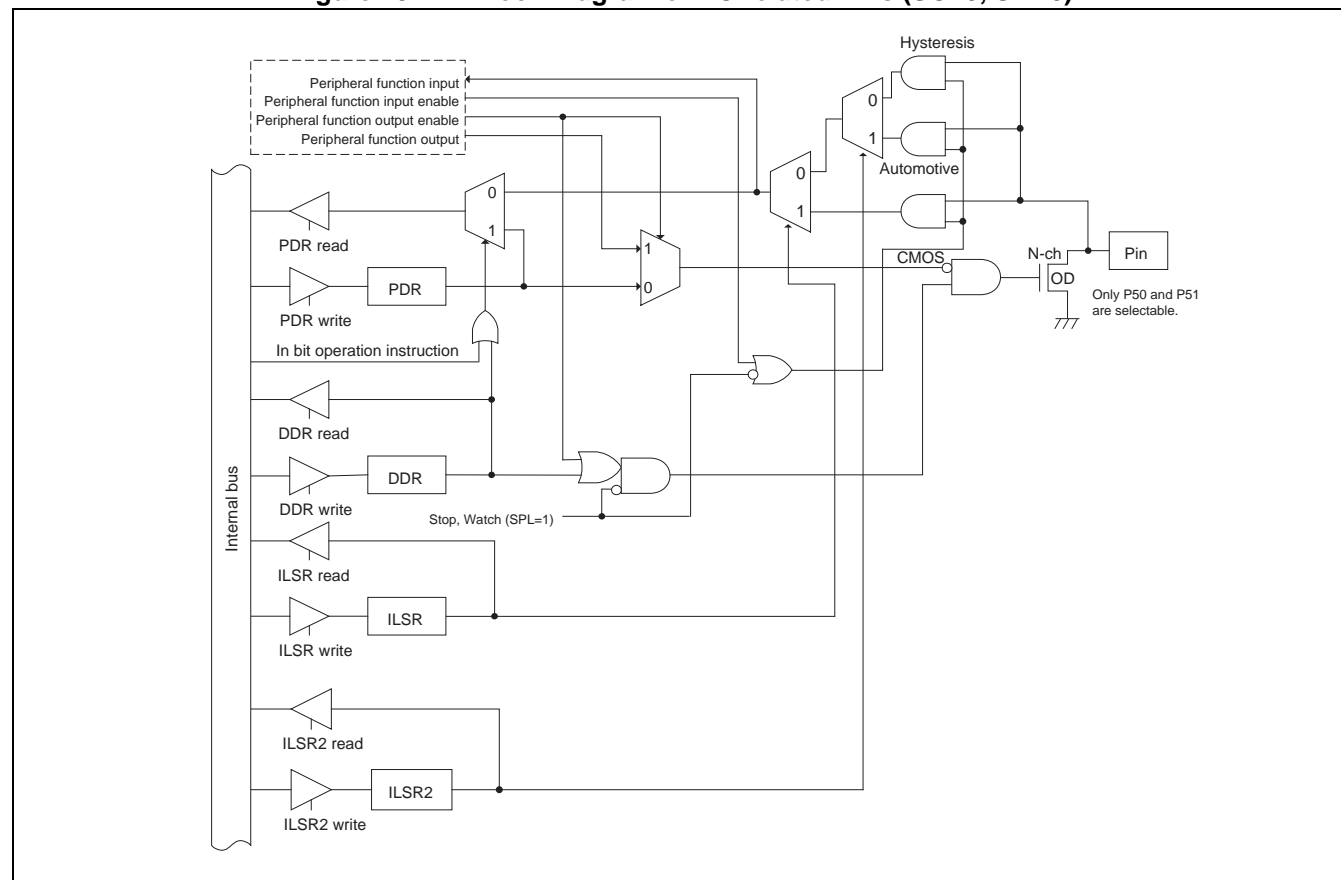
The SCL0 pin can serve as a N-ch open-drain I/O port, external interrupt input (hysteresis input), serial data input (hysteresis input) for eight-bit serial I/O, or I<sup>2</sup>C serial clock I/O pin (SCL0).

SCL0: When I<sup>2</sup>C is enabled (ICCR0:EN = 1), the SCL0 pin is automatically set as the shift clock I/O pin to function as the SCL0 pin.

To use it as an input pin, enable the I<sup>2</sup>C operation (ICCR0: EN = 1) and write "0" to the corresponding of bit4 port direction register (DDR).

## ■ I<sup>2</sup>C-related Pin Block Diagram

Figure 23.4-1 Block Diagram of I<sup>2</sup>C-related Pins (SCL0, SDA0)



23.5 I<sup>2</sup>C Registers

This section describes the I<sup>2</sup>C registers.

■ I<sup>2</sup>C RegistersFigure 23.5-1 I<sup>2</sup>C RegistersI<sup>2</sup>C bus control register 0 (IBCR00)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0060 <sub>H</sub> IBCR00	AACKX	INTS	ALF	ALE	SPF	SPE	WUF	WUE	00000000 <sub>B</sub>
	R/W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	R(RM1),W	R/W	

I<sup>2</sup>C bus control register 1 (IBCR10)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0061 <sub>H</sub> IBCR10	BER	BEIE	SCC	MSS	DACKE	GACKE	INTE	INT	00000000 <sub>B</sub>
	R(RM1),W	R/W	R0,W	R/W	R/W	R/W	R/W	R(RM1),W	

I<sup>2</sup>C bus status register (IBSR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0062 <sub>H</sub> IBSR0	BB	RSC	-	LRB	TRX	AAS	GCA	FBT	00000000 <sub>B</sub>
	R/WX	R/WX	R0,WX	R/WX	R/WX	R/WX	R/WX	R/WX	

I<sup>2</sup>C data register (IDDR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0063 <sub>H</sub> IDDR0	D7	D6	D5	D4	D3	D2	D1	D0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

I<sup>2</sup>C address register (IAAR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064 <sub>H</sub> IAAR0	-	A6	A5	A4	A3	A2	A1	A0	00000000 <sub>B</sub>
	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

I<sup>2</sup>C clock control register (ICCR0)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0065 <sub>H</sub> ICCR0	DMBP	-	EN	CS4	CS3	CS2	CS1	CS0	00000000 <sub>B</sub>
	R/W	R0,WX	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (Read value is the same as write value)

R(RM1),W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

R0,W : Write only (Writable, "0" is read)

R/WX : Read only (Readable, writing has no effect on operation)

R0,WX : Undefined bit (Read value is "0", writing has no effect on operation)

## 23.5.1 I<sup>2</sup>C Bus Control Registers 0, 1 (IBCR00, IBCR10)

The I<sup>2</sup>C bus control registers are used to select the operating mode and to enable or disable interrupts, acknowledgment, general call acknowledgment, and MCU standby wakeup function.

### ■ I<sup>2</sup>C Bus Control Register 0 (IBCR00)

Figure 23.5-2 I<sup>2</sup>C Bus Control Register 0 (IBCR00)

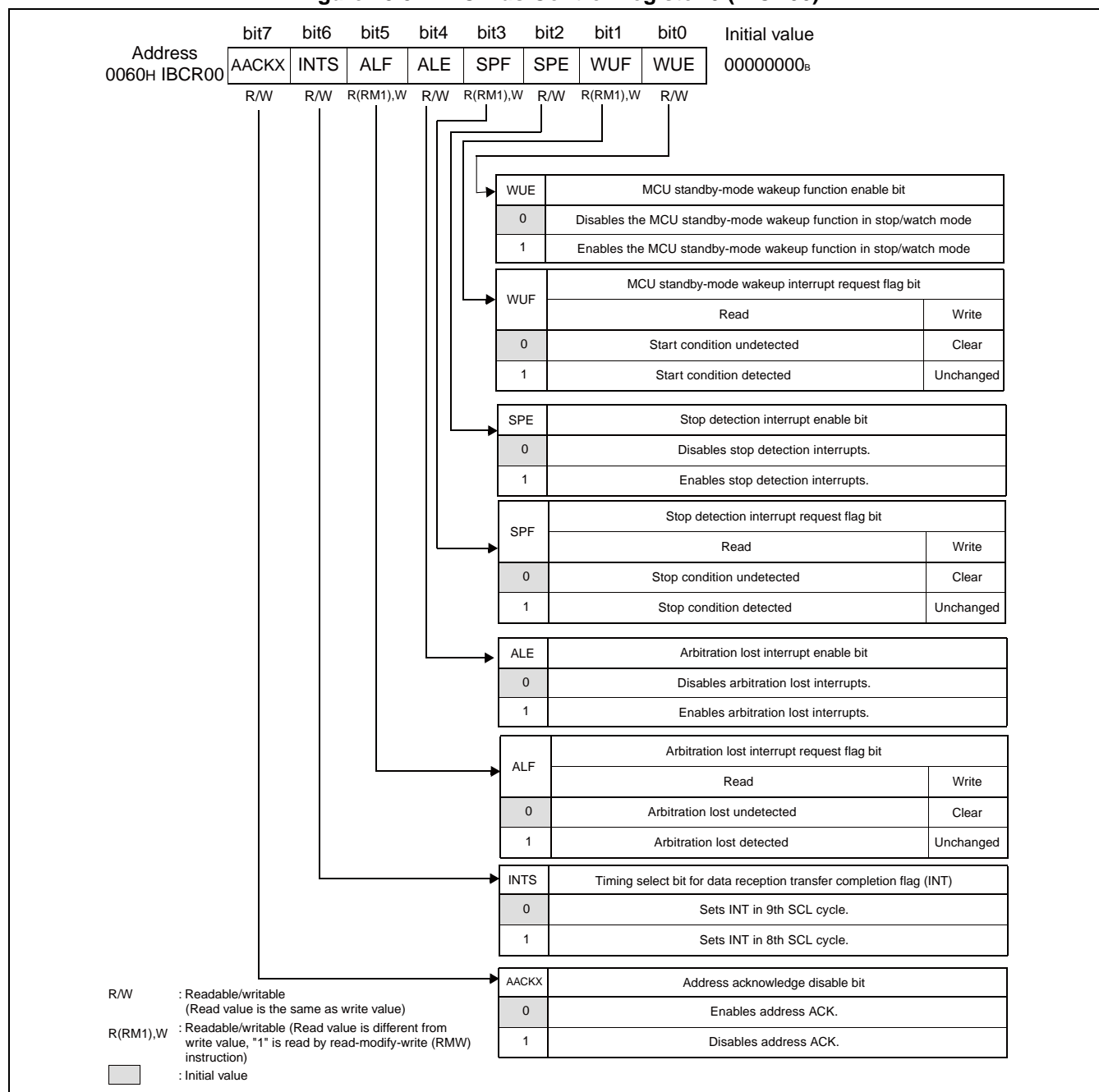


Table 23.5-1 I<sup>2</sup>C Bus Control Register 0 (IBCR00) (1 / 2)

Bit name		Function
bit7	AACKX: Address acknowledge disable bit	<p>This bit controls the address ACK when the first byte is transmitted.</p> <p><b>Setting the bit to "0":</b> Causes the address ACK to be output automatically (The address ACK is returned automatically if the slave address matches).</p> <p><b>Setting the bit to "1":</b> Prevents the address ACK from being output.</p> <p>Update this bit in either of the following ways:</p> <ul style="list-style-type: none"> <li>- Write "1" to the bit in master mode.</li> <li>- Clear the bit to "0" after making sure that the bus busy bit is "0" (IBSR0:BB = 0).</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>• If AACKX = 1 and IBSR0:FBT = 0 when an IBCR10:INT bit interrupt occurs, no address ACK is output even though the I<sup>2</sup>C address matches the slave address. Clear the IBCR10:INT bit to "0" as an interrupt is generated upon completion of transfer of each byte of address/data in the same way as during addressing.</li> <li>• If AACKX = 1 and IBSR0:FBT = 1 when an IBCR10:INT bit interrupt occurs, "1" might be written to AACKX after addressing as in slave mode. Either continue normal communication after setting AACKX to "0" again or restart communication after disabling I<sup>2</sup>C operation (ICCR0:EN = 0).</li> </ul>
bit6	INTS: Timing select bit for data reception transfer completion flag (INT)	<p>This bit selects the timing of the transfer completion interrupt (IBCR10:INT) when data is received. Change the bit only when IBSR0:TRX = 0 and IBSR0:FBT = 0.</p> <p><b>Setting the bit to "0":</b> Sets the transfer completion interrupt (IBCR10:INT) in the ninth SCL cycle.</p> <p><b>Setting the bit to "1":</b> Sets the transfer completion interrupt (IBCR10:INT) in the eighth SCL cycle.</p> <p>Notes:</p> <ul style="list-style-type: none"> <li>• The transfer completion interrupt (IBCR0:INT) is set always in the ninth SCL0 cycle except during data reception (IBSR0:TRX = 1 or IBSR0:FBT = 1).</li> <li>• If the data ACK depends on the content of the received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACE) after writing "1" to this bit (for example, using a previous transfer completion interrupt) to read latest received data.</li> <li>• The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt in the ninth SCL cycle.) If ACK is read when this bit is "1", therefore, you must write "0" to this bit in the transfer completion interrupt in the eighth SCL0 cycle so that another transfer completion interrupt will occur in the ninth SCL0 cycle.</li> </ul>
bit5	ALF: Arbitration lost interrupt request flag bit	<p>This bit is used to detect when arbitration is lost.</p> <ul style="list-style-type: none"> <li>• An arbitration lost interrupt request is generated if this bit and the IBCR00:ALE bit are both "1".</li> <li>• This bit is set to "1" in the following cases: <ul style="list-style-type: none"> <li>- When arbitration lost is detected during data/address transmission as a master</li> <li>- When "1" is written to the IBCR10:MSS bit with the bus being used by another system. However, the bit is not set when "1" is written to the MSS bit after the system returns AACK or GACK as a slave.</li> </ul> </li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- When "0" is written to the IBCR00:ALF bit with IBSR0:BB = 0.</li> <li>- When "0" is written to the IBCR10:INT bit to clear the transmission completion flag.</li> </ul> </li> <li>• Writing "1" to this bit leaves its value unchanged and has no effect on the operation.</li> <li>• The bit returns "1" when read by a read-modify-write (RMW) instruction.</li> </ul>
bit4	ALE: Arbitration lost interrupt enable bit	<p>This bit enables or disables arbitration lost interrupts.</p> <p>An arbitration lost interrupt request is generated if this bit and the IBCR00:ALF bit are both "1".</p> <p><b>Setting the bit to "0":</b> Disables arbitration lost interrupts.</p> <p><b>Setting the bit to "1":</b> Enables arbitration lost interrupts.</p>
bit3	SPE: Stop detection interrupt request flag bit	<p>This bit is used to detect a stop condition.</p> <ul style="list-style-type: none"> <li>• A stop detection interrupt request is generated if this bit and the IBCR00:SPE bit are both "1".</li> <li>• This bit is set to "1" if a valid stop condition is detected when the bus is busy.</li> </ul> <p><b>Setting the bit to "0":</b> Clears itself (changes the value to "0").</p> <p><b>Setting the bit to "1":</b> Leaves its value unchanged without affecting the operation.</p> <ul style="list-style-type: none"> <li>• The bit returns "1" when read by a read-modify-write (RMW) instruction.</li> </ul>



**Table 23.5-1 I<sup>2</sup>C Bus Control Register 0 (IBCR00) (2 / 2)**

Bit name		Function
bit2	SPE: Stop detection interrupt enable bit	This bit enables or disables stop detection interrupts. A stop detection interrupt request is generated if this bit and the IBCR00:SPF bit are both "1". <b>Setting the bit to "0"</b> : Disables stop detection interrupts. <b>Setting the bit to "1"</b> : Enables stop detection interrupts.
bit1	WUF: MCU standby-mode wakeup interrupt request flag bit	This bit is used to detect MCU wakeup from a standby mode (stop or watch mode). <ul style="list-style-type: none"> <li>A wakeup interrupt request is generated if this bit and the IBCR00:WUE bit are both "1".</li> <li>This bit is set to "1" if a start condition is detected with the wakeup function enabled (IBCR00:WUE = 1).</li> </ul> <b>Setting the bit to "0"</b> : Clears itself (changes the value to "0"). <b>Setting the bit to "1"</b> : Leaves its value unchanged without affecting the operation. <ul style="list-style-type: none"> <li>The bit returns "1" when read by a read-modify-write (RMW) instruction.</li> </ul>
bit0	WUE: MCU standby-mode wakeup function enable bit	This bit enables or disables the function to wake up the MCU from standby mode (stop or watch mode). <b>Setting the bit to "0"</b> : Disables the wakeup function. <b>Setting the bit to "1"</b> : Enables the wakeup function. If a start condition is detected in stop or watch mode when this bit is "1", a wakeup interrupt request is generated to start I <sup>2</sup> C operation.  Notes: <ul style="list-style-type: none"> <li>Write "1" to this bit immediately before the MCU enters the stop or watch mode. To ensure that I<sup>2</sup>C operation can restart immediately after the MCU wakes up from stop or watch mode, clear (write "0" to) this bit as soon as possible.</li> <li>When a wakeup interrupt request occurs, the MCU wakes up after the oscillation stabilization wait time elapses. To prevent the data loss immediately after wakeup, therefore, the SCL0 must rise as the first cycle and the first bit must be received as data after 100 ms (assuming that the minimum oscillation stabilization wait time is 100 ms) from the wakeup due to the start of I<sup>2</sup>C transmission (upon detection of the falling edge of SDA0).</li> <li>During a MCU standby mode, the status flags, state machine, and I<sup>2</sup>C bus outputs for the I<sup>2</sup>C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I<sup>2</sup>C bus system, make sure that IBSR0:BB = 0 before entering standby mode.</li> <li>The wakeup function does not support the transition of the MCU to stop or watch mode with IBSR0:BB = 1. If the MCU enters stop or watch mode with IBSR0:BB = 1, a bus error will occur upon detection of a start condition.</li> <li>The wakeup function is useful only when the MCU remains in stop/watch mode. (In PLL stop mode, for example, the time from wakeup to the start of communication becomes longer than in stop/watch mode as the PLL oscillation stabilization wait time is required in addition to the oscillation stabilization wait time.)</li> </ul>

Note: The AACKX, INTS, and WUE bits in the IBCR00 register are set to "0" and cannot be written to either when I<sup>2</sup>C operation is disabled (ICCR:EN = 0) or when a bus error occurs (IBCR10:BER = 1).

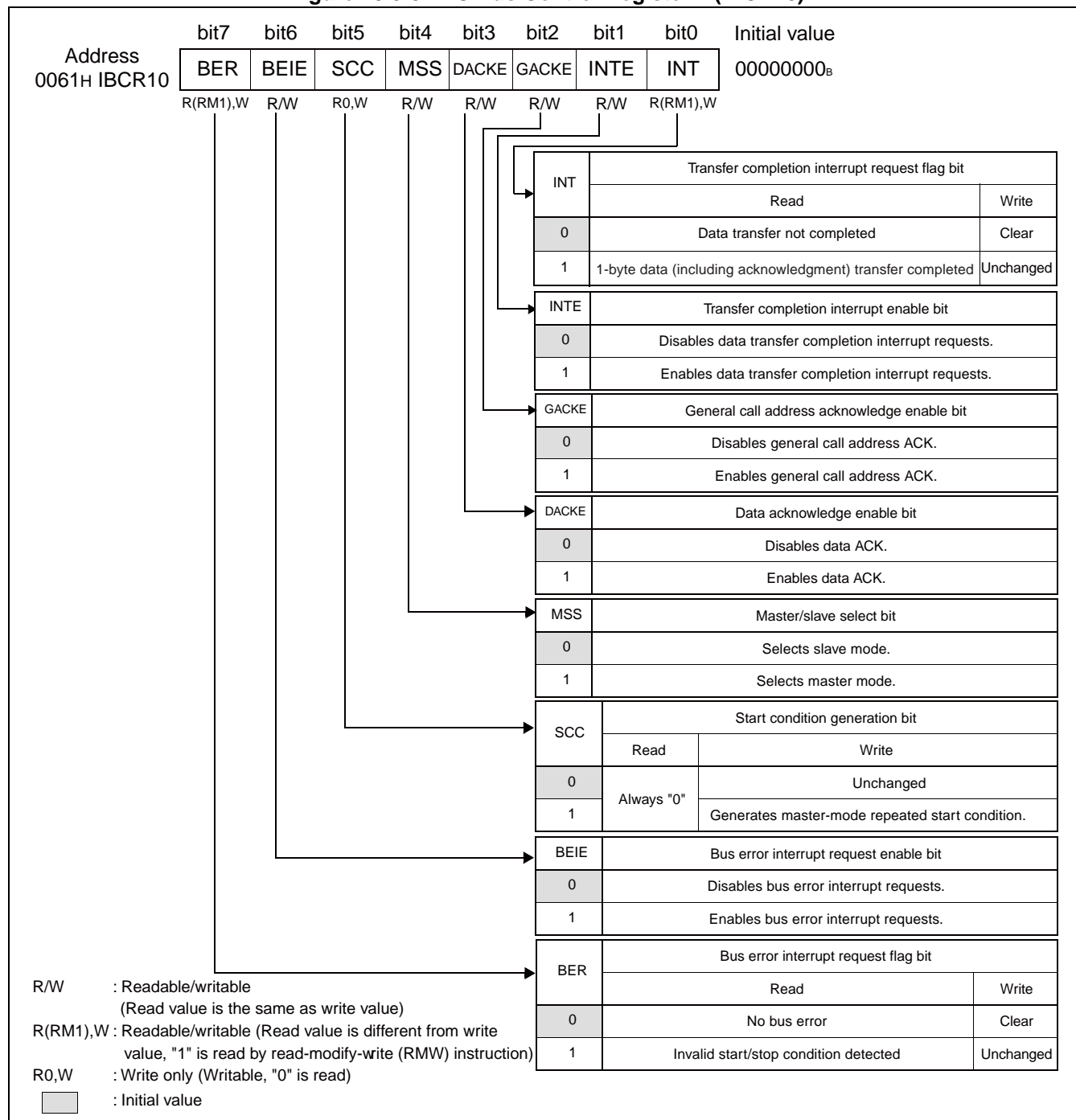
■ I<sup>2</sup>C Bus Control Register 1 (IBCR10)Figure 23.5-3 I<sup>2</sup>C Bus Control Register 1 (IBCR10)

Table 23.5-2 I<sup>2</sup>C Bus Control Register 1 (IBCR10) (1 / 2)

Bit name		Function
bit7	BER: Bus error interrupt request flag bit	<p>This bit is used to detect bus errors.</p> <ul style="list-style-type: none"> <li>A bus error interrupt request is generated if this bit and the IBCR10:BEIE bit are both "1".</li> <li>This bit is set to "1" when an invalid start or stop condition is detected.</li> </ul> <p><b>Setting the bit to "0":</b> Clears itself (changes the value to "0").</p> <p><b>Setting the bit to "1":</b> Leaves its value unchanged without affecting the operation.</p> <ul style="list-style-type: none"> <li>The bit returns "1" when read by a read-modify-write (RMW) instruction.</li> <li>When this bit is set to "1", ICCR0:EN is set to "0", and the I<sup>2</sup>C interface enters halt mode to terminate data transfer.</li> </ul>
bit6	BEIE: Bus error interrupt request enable bit	<p>This bit enables or disables bus error interrupts.</p> <p>A bus error interrupt request is generated if this bit and the IBCR10:BER bit are both "1".</p> <p><b>Setting the bit to "0":</b> Disables bus error interrupts.</p> <p><b>Setting the bit to "1":</b> Enables bus error interrupts.</p>
bit5	SCC: Start condition generation bit	<p>This bit can be used to generate a start condition repeatedly to restart communications in master mode.</p> <ul style="list-style-type: none"> <li>Writing "1" to the bit in master mode generates a start condition repeatedly.</li> <li>Writing "0" to the bit is meaningless.</li> <li>When read, the bit returns "0".</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>Do not set IBCR10:SCC = 1 and IBCR10:MSS = 0 at the same time.</li> <li>An attempt to write "1" to this bit is ignored when IBCR10:INT = 0 (no start condition is generated). If you write "1" to this bit and "0" to the IBCR10:INT bit at the same time when the IBCR10:INT = 1, this bit takes priority and generates a start condition.</li> </ul>
bit4	MSS: Master/slave select bit	<p>This bit selects master mode or slave mode.</p> <ul style="list-style-type: none"> <li>Writing "1" to this bit while the I<sup>2</sup>C bus is in the idle state (IBSR0:BB = 0) selects master mode, generates a start condition, and then starts address transfer.</li> <li>Writing "0" to the bit while the I<sup>2</sup>C bus is in the busy state (IBSR0:BB = 1) selects slave mode, generates a stop condition, and then ends data transfer.</li> <li>If arbitration lost occurs during data or address transfer in master mode, this bit is cleared to "0" and the mode changes to slave mode.</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>Do not set IBCR10:SCC = 1 and IBCR10:MSS = 0 at the same time.</li> <li>An attempt to write "0" to this bit is ignored when IBCR10:INT = 0. If you write "0" to this bit and "0" to the IBCR10:INT bit at the same time when the IBCR10:INT = 1, this bit takes priority and generates a stop condition.</li> <li>The IBCR00:ALF bit is not set even though you write "1" to the MSS bit during transmission or reception in slave mode. Do not write "1" to the MSS bit during transmission or reception in slave mode.</li> </ul>
bit3	DACKE: Data acknowledge enable bit	<p>This bit controls data acknowledgment during data reception.</p> <p><b>Setting the bit to "0":</b> Disables data acknowledge output.</p> <p><b>Setting the bit to "1":</b> Enables data acknowledge output. In this case, data acknowledgment is output in the ninth SCL0 cycle during data reception in master mode. In slave mode, data acknowledgment is output in the ninth SCL0 cycle only if address acknowledgment has already been output.</p>
bit2	GACKE: General call address acknowledge enable bit	<p>This bit controls general call address acknowledgment.</p> <p><b>Setting the bit to "0":</b> Disables output of general call address acknowledge.</p> <p><b>Setting the bit to "1":</b> Causes a general call address acknowledgment to be output if a general call address (00H) is received in master or slave mode.</p>
bit1	INTE: Transfer completion interrupt enable bit	<p>This bit enables or disables transfer completion interrupts.</p> <p><b>Setting the bit to "0":</b> Disables transfer completion interrupts.</p> <p><b>Setting the bit to "1":</b> Enables transfer completion interrupts.</p> <p>A transfer completion interrupt request is generated if this bit and the IBCR10:INT bit are both "1".</p>

**Table 23.5-2 I<sup>2</sup>C Bus Control Register 1 (IBCR10) (2 / 2)**

Bit name		Function
bit0	INT: Transfer completion interrupt request flag bit	<p>This bit is used to detect transfer completion.</p> <ul style="list-style-type: none"> <li>A transfer completion interrupt request is generated if this bit and the IBCR10:INTE bit are both "1".</li> <li>This bit is set to "1" upon completion of transfer of 1-byte address or data (whether or not this includes an acknowledgment depends on the IBCR00:INTS setting) if any of the following four conditions is satisfied. <ul style="list-style-type: none"> <li>In bus master mode</li> <li>Addressed as slave</li> <li>General call address received</li> <li>Arbitration lost detected</li> </ul> </li> <li>This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>"0" written to the bit</li> <li>Repeated start condition (IBCR10:SCC = 1) or stop condition (IBCR10:MSS = 0) occurred in master mode.</li> </ul> </li> <li>An attempt to write "1" to this bit leaves its value unchanged and has no effect on the operation.</li> <li>The bit returns "1" when read by a read-modify-write (RMW) instruction.</li> <li>The SCL0 line remains at "L" while this bit is "1".</li> <li>Writing "0" to clear the bit (change the value to "0") releases the SCL0 line to enable transmission for the next byte of data.</li> </ul> <p>Notes:</p> <ul style="list-style-type: none"> <li>If "1" is written to IBCR10:SCC when this bit is "0", the IBCR10:SCC bit has priority and the start condition is generated.</li> <li>If "0" is written to IBCR10:MSS when this bit is "0", the IBCR10:MSS bit has priority and the stop condition is generated.</li> <li>If IBCR00:INTS = 1 when data is received, this bit is set to "1" upon completion of transfer of one-byte data (including no acknowledgment). In other cases, this bit is set to "1" upon completion of transmission or reception of one-byte data/address including an acknowledgment.</li> </ul>

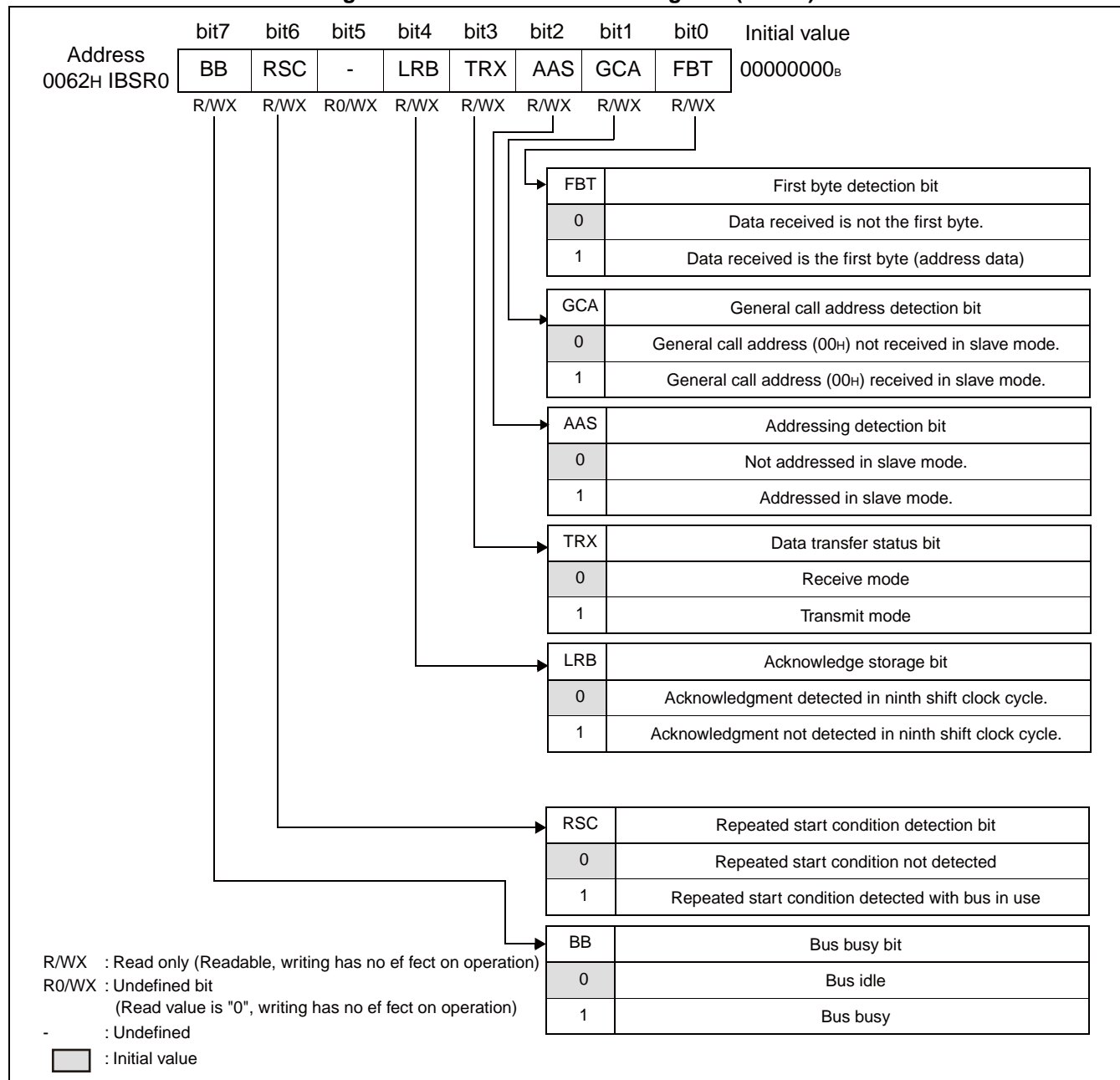
- Notes:
- When clearing the interrupt request flag (IBCR10:BER) by writing "0", do not update the interrupt request enable bit (IBCR10:BEIE) at the same time.
  - All the bits in IBCR10 except the BER and BEIE bits are cleared to "0" either when operation is disabled (ICCR:EN = 0) or when a bus error occurs (IBCR10:BER = 1).

## 23.5.2 I<sup>2</sup>C Bus Status Register (IBSR0)

The IBSR0 register contains the status of the I<sup>2</sup>C interface.

### ■ I<sup>2</sup>C Bus Status Register (IBSR0)

Figure 23.5-4 I<sup>2</sup>C Bus Status Register (IBSR0)



**Table 23.5-3 I<sup>2</sup>C Bus Status Register (IBSR0)**

Bit name		Function
bit7	BB: Bus busy bit	This bit indicates the bus status. <ul style="list-style-type: none"> <li>• This bit is set to "1" when a start condition is detected.</li> <li>• This bit is set to "0" when a stop condition is detected.</li> </ul>
bit6	RSC: Repeated start condition detection bit	This bit is used to detect repeated start conditions. <ul style="list-style-type: none"> <li>• This bit is set to "1" when a repeated start condition is detected.</li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- When "0" is written to IBCR10:INT.</li> <li>- When the slave address does not match the address set in IAAR0 in slave mode.</li> <li>- When the slave address matches the address set in IAAR0 but IBCR00:AACKX = 1 in slave mode.</li> <li>- When the general call address is received but IBCR10:GACKE = 0 in slave mode.</li> <li>- When a stop condition is detected.</li> </ul> </li> </ul>
bit5	Undefined bit	The value read is always "0". An attempt to write to the bit is meaningless.
bit4	LRB: Acknowledge storage bit	This bit saves the value of the SDA0 line in the ninth shift clock cycle during data byte transfer. <ul style="list-style-type: none"> <li>• This bit is set to "1" when no acknowledgment is detected (SDA0 = "H").</li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- When acknowledgment is detected (SDA0 = "L")</li> <li>- When a start or stop condition is detected.</li> </ul> </li> </ul> <p>Note: It follows from the above that this bit must be read after ACK (Read the value in response to the transfer completion interrupt in the ninth SCL0 cycle). Accordingly, if ACK is read when the IBCR00:INTS bit is "1", you must write "0" to the IBCR00:INTS bit in the transfer completion interrupt triggered by the eighth SCL0 cycle so that another transfer completion interrupt will be triggered by the ninth SCL0 cycle.</p>
bit3	TRX: Data transfer status bit	This bit indicates the data transfer mode. <ul style="list-style-type: none"> <li>• This bit is set to "1" when data transfer is performed in transfer mode.</li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- Data is transferred in receive mode.</li> <li>- NACK is received in slave transmit mode.</li> </ul> </li> </ul>
bit2	AAS: Addressing detection bit	This bit indicates that the MCU has been addressed in slave mode. <ul style="list-style-type: none"> <li>• This bit is set to "1" if the MCU is addressed in slave mode.</li> <li>• This bit is set to "0" when a start or stop condition is detected.</li> </ul>
bit1	GCA: General call address detection bit	This bit is used to detect a general call address. <ul style="list-style-type: none"> <li>• This bit is set to "1" in the following cases: <ul style="list-style-type: none"> <li>- When the general call address (00<sub>H</sub>) is received in slave mode.</li> <li>- When the general call address (00<sub>H</sub>) is received in master mode with IBCR10:GACKE = 1.</li> <li>- When arbitration lost is detected during transmission of the second byte of the general call address in master mode.</li> </ul> </li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- When a start or stop condition is detected.</li> <li>- When arbitration lost is not detected during transmission of the second byte of the general call address in master mode.</li> </ul> </li> </ul>
bit0	FBT: First byte detection bit	This bit is used to detect first byte. <ul style="list-style-type: none"> <li>• This bit is set to "1" when a start condition is detected.</li> <li>• This bit is set to "0" in the following cases: <ul style="list-style-type: none"> <li>- When "0" is written to the IBCR10:INT bit.</li> <li>- When the slave address does not match the address set in IAAR0 in slave mode.</li> <li>- When the slave address matches the address set in IAAR0 but IBCR00:AACKX = 1 in slave mode.</li> <li>- When the general call address is received with IBCR10:GACKE = 0 in slave mode.</li> </ul> </li> </ul>

23.5.3 I<sup>2</sup>C Data Register (IDDR0)

The IDDR0 register is used to set the data or address to send and to hold the data or address received.

■ I<sup>2</sup>C Data Register (IDDR0)

Figure 23.5-5 I<sup>2</sup>C Data Register (IDDR0)

I <sup>2</sup> C data register (IDDR0)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0063 <sub>H</sub> IDDR0	D7	D6	D5	D4	D3	D2	D1	D0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W: Readable/writable (Read value is the same as write value)									

In transmit mode, each bit of the data or address value written to the register is shifted to the SDA0 line, starting with the MSB. The write side of this register is double-buffered, where if the bus is in use (IBSR0:BB=1), the write data is loaded to the 8-bit shift register either when the current data transfer completion interrupt is cleared (writing "0" to the IBCR10:INT bit) or when a repeated start condition is generated (writing "1" to the IBCR10:SCC bit). Each bit of the shift register data is output (shifted) to the SDA0 line.

Note that writing to this register has no effect on the current data transfer. In slave mode, however, data is transferred to the shift register after the address is determined.

The received data or address can be read from this register during the transfer completion interrupt (IBCR10:INT = 1). When it is read, however, the serial transfer register is directly read from, the receive data is valid only while IBCR10:INT = 1.

## 23.5.4 I<sup>2</sup>C Address Register (IAAR0)

The IAAR0 register is used to set the slave address.

### ■ I<sup>2</sup>C Address Register (IAAR0)

Figure 23.5-6 I<sup>2</sup>C Address Register (IAAR0)

I <sup>2</sup> C address register (IAAR0)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0064 <sub>H</sub> IAAR0	-	A6	A5	A4	A3	A2	A1	A0	00000000 <sub>B</sub>
	R0/WX	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : Readable/writable (Read value is the same as write value)									
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)									
- : Undefined									

The I<sup>2</sup>C address register (IAAR0) is used to set the slave address. In slave mode, address data from the master is received and then compared with the value of the IAAR register.



23.5.5 I<sup>2</sup>C Clock Control Register (ICCR0)

The ICCR0 register is used to enable I<sup>2</sup>C operation and select the shift clock frequency.

■ I<sup>2</sup>C Clock Control Register (ICCR0)

Figure 23.5-7 I<sup>2</sup>C Clock Control Register (ICCR0)

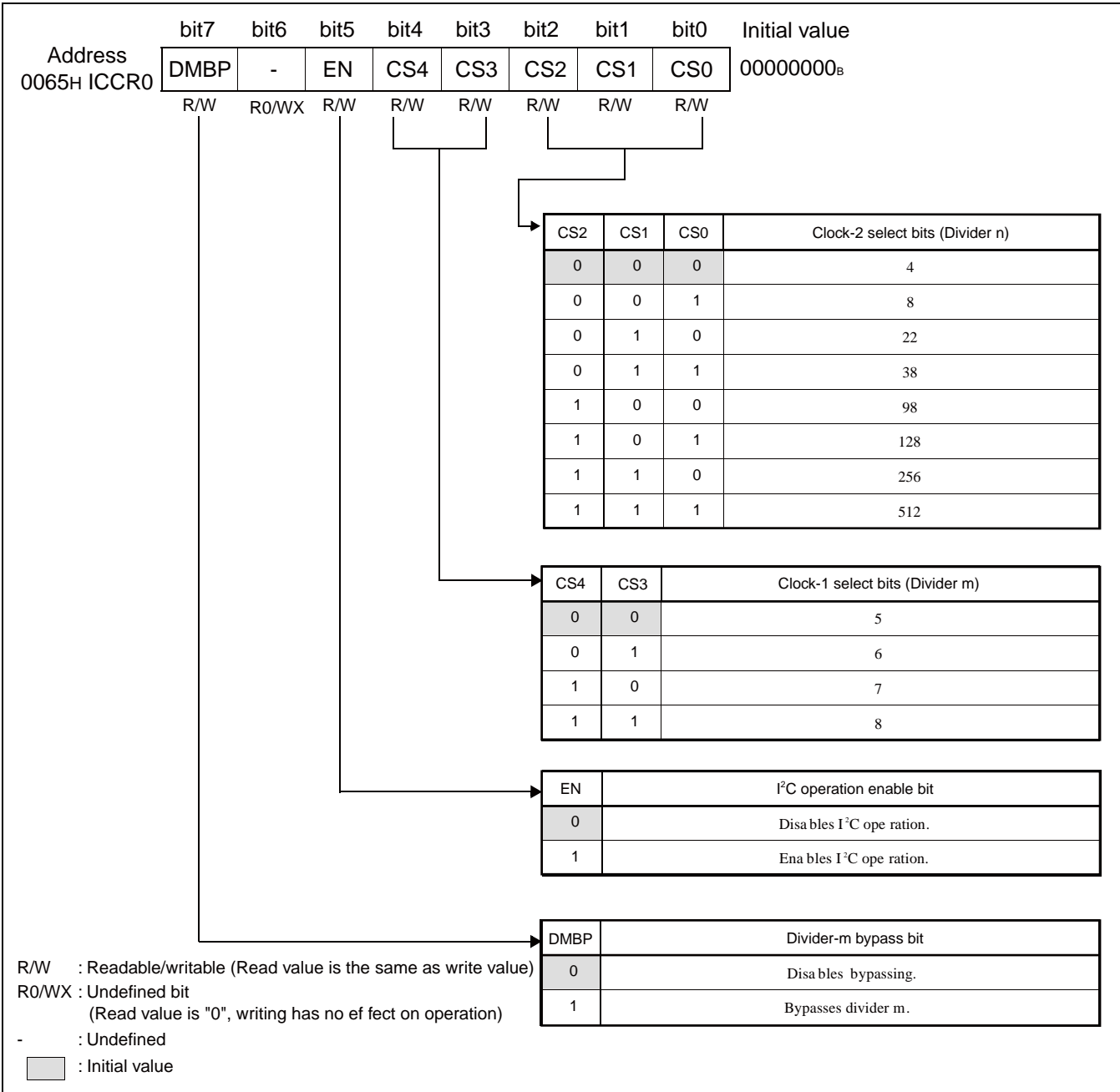


Table 23.5-4 I<sup>2</sup>C Clock Control Register (ICCR0)

Bit name		Function
bit7	DMBP: Divider-m bypass bit	This bit is used to bypass the divider m to generate the shift clock frequency. <b>Setting the bit to "0"</b> : Sets the value set in CS3 and CS4 as the divider m value (m = ICCR0:CS4, 3). <b>Setting the bit to "1"</b> : Bypasses the divider m. Note: Do not set this bit to "1" when divider n = 4 (ICCR0:CS2 to CS0 = 000 <sub>B</sub> ).
bit6	Undefined bit	The value read is always "0". An attempt to write to the bit is meaningless.
bit5	EN: I <sup>2</sup> C operation enable bit	<ul style="list-style-type: none"> <li>This bit enables I<sup>2</sup>C interface operation.</li> </ul> <b>Setting the bit to "0"</b> : Disables operation of the I <sup>2</sup> C interface and clears the following bits to "0". <ul style="list-style-type: none"> <li>- AACKX, INTS, and WUE bits in the IBCR00 register</li> <li>- All the bits in the IBCR10 register except the BER and BEIE bits</li> <li>- All bits in the IBSR0 register</li> </ul> <b>Setting the bit to "1"</b> : Enables operation of the I <sup>2</sup> C interface. <ul style="list-style-type: none"> <li>This bit is set to "0" in the following cases:               <ul style="list-style-type: none"> <li>- When "0" is written to this bit.</li> <li>- When IBCR10:BER is "1".</li> </ul> </li> </ul>
bit4, bit3	CS4, CS3: Clock-1 select bits (Divider m)	These bits set the shift clock frequency. Shift clock frequency (F <sub>sck</sub> ) is set as shown by the following equation: $F_{sck} = \frac{\phi}{(m \times n + 2)}$ φ represents the machine clock frequency (MCLK).
bit2, bit1, bit0	CS2, CS1, CS0: Clock-2 select bits (Divider n)	

Note: If the standby mode wakeup function is not used, disable I<sup>2</sup>C operation before switching the MCU to stop or watch mode.

## 23.6 I<sup>2</sup>C Interrupts

The I<sup>2</sup>C interface has a transfer interrupt and a stop interrupt which are triggered by the following events.

- **Transfer interrupt**

A transfer interrupt occurs either upon completion of data transfer or when a bus error occurs.

- **Stop interrupt**

A stop interrupt occurs upon detection of a stop condition or arbitration lost or upon access to the I<sup>2</sup>C interface in stop/watch mode.

### ■ Transfer Interrupt

Table 23.6-1 shows the transfer interrupt control bits and I<sup>2</sup>C interrupt sources.

**Table 23.6-1 Transfer Interrupt Control Bits and I<sup>2</sup>C Interrupt Sources**

Item	End of transfer	Bus error
Interrupt request flag bit	IBCR10:INT =1	IBCR10:BER =1
Interrupt request enable bit	IBCR10:INTE =1	IBCR10:BEIE =1
Interrupt source	Data transfer complete	Bus error occurred

- **Interrupt upon completion of transfer**

An interrupt request is output to the CPU upon completion of data transfer if the transfer completion interrupt request enable bit has been set to enable (IBCR10:INTE = 1). In the interrupt service routine, write "0" to the transfer completion interrupt request flag bit (IBCR10:INT) to clear the interrupt request. When data transfer is completed, the IBCR10:INT bit is set to "1" regardless of the value of the IBCR10:INTE bit.

- **Interrupt in response to a bus error**

When the following conditions are met, a bus error is deemed to have occurred, and the I<sup>2</sup>C interface will be stopped.

- When a stop condition is detected in master mode.
- When a start or stop condition is detected during transmission or reception of the first byte.
- When a start or stop condition is detected during transmission or reception of data (excluding the start, first data, and stop bits).

In these cases, an interrupt request is output to the CPU if the bus error interrupt request enable bit has been set to enable (IBCR10:BEIE = 1). In the interrupt service routine, write "0" to the bus error interrupt request flag bit (IBCR10:BER) to clear the interrupt request. When a bus error occurs, the IBCR10:BER bit is set to "1" regardless of the value of the IBCR10:BEIE bit.

## ■ Stop Interrupt

Table 23.6-2 shows the stop interrupt control bits and I<sup>2</sup>C interrupt sources (trigger events).

**Table 23.6-2 Stop Interrupt Control Bits and I<sup>2</sup>C Interrupt Sources**

Item	Detection of stop condition	Detection of arbitration lost	MCU wakeup from stop/watch mode
Interrupt request flag bit	IBCR00:SPF =1	IBCR00:ALF =1	IBCH00:WUF =1
Interrupt request enable bit	IBCR00:SPE =1	IBCR00:ALE =1	IBCR00:WUE =1
Interrupt source	Stop condition detected	Arbitration lost detected	Start condition detected

- Interrupt upon detection of a stop condition

A stop condition is considered to be valid if all of the following conditions are satisfied when the stop condition is detected.

- The bus is busy (state which the start condition is detected).
- IBCR10:MSS = 0
- After transfer of one byte of data completes, including the acknowledgment.

In this case, an interrupt request is output to the CPU if the stop condition detection interrupt request enable bit has been set to enable (IBCR00:SPE =1). In the interrupt service routine, write "0" to the IBCR00:SPF bit to clear the interrupt request.

The IBCR00:SPF bit is set to "1" when a valid stop condition occurs regardless of the value of the IBCR00:SPE bit.

- Interrupt upon detection of arbitration lost

When arbitration lost is detected, an interrupt request is output to the CPU if the arbitration lost detection interrupt request enable bit has been set to enable (IBCR00:ALE = 1). Either write "0" to the arbitration lost interrupt request flag bit (IBCR00:ALF) while the bus is idle or write "0" to the IBCR10:INT bit from the interrupt service routine while the bus is busy to clear the interrupt request.

When arbitration lost occurs, the IBCR00:ALF bit is set to "1" regardless of the value for the IBCR00:ALE bit.

- Interrupt for MCU wakeup from stop/watch mode

When a start condition is detected, an interrupt request is output to the CPU if the function to wake up the MCU from stop or watch mode has been enabled (IBCR00:WUE = 1).

In the interrupt service routine, write "0" to the MCU standby mode wakeup interrupt request flag bit (IBCR00:WUF) to clear the interrupt request.

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt source numbers and vector tables of all peripheral functions.

■ Registers and Vector Table Related to I<sup>2</sup>C Interrupts

Table 23.6-3 Registers and Vector Table Related to I<sup>2</sup>C Interrupts

Channel	Interrupt request No.	Interrupt Level Setting register		Vector table address	
		Register	Setting bit	Upper	Lower
ch.0	IRQ16	ILR4	L16	FFDA <sub>H</sub>	FFDB <sub>H</sub>

ch.: channel

## 23.7 I<sup>2</sup>C Operations and Setup Procedure Examples

---

This section describes the operation of I<sup>2</sup>C.

---

### ■ Operation of I<sup>2</sup>C

#### ● I<sup>2</sup>C interface

The I<sup>2</sup>C interface is an eight-bit serial interface synchronized with a shift clock. It conforms to the I<sup>2</sup>C bus specification defined by Philips.

#### ● MCU standby mode wakeup function

The wakeup function wakes up the MCU upon detection of a start condition, from low power consumption mode such as stop or watch mode.

### ■ Setup Procedure Example

Use the following procedure to set up I<sup>2</sup>C:

#### ● Initial setting

- 1) Set the port for input (DDR0).
- 2) Set the interrupt level (ILR2, ILR4).
- 3) Set the slave address (IAAR0).
- 4) Select the clock and enable I<sup>2</sup>C operation (ICCR0).
- 5) Enable bus error interrupt requests (IBCR00:BEIE = 1).

#### ● Interrupt processing

- 1) Arbitrary processing
- 2) Clear the bus error interrupt request flag (IBCR00:BER = 0).

## 23.7.1 I<sup>2</sup>C Interface

The I<sup>2</sup>C interface is an eight-bit serial interface synchronized with the shift clock. It conforms to the I<sup>2</sup>C bus specification defined by Philips.

### ■ I<sup>2</sup>C System

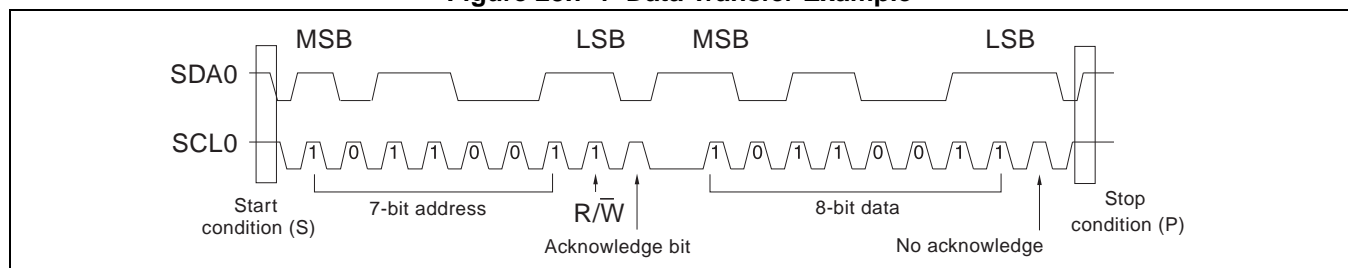
The I<sup>2</sup>C bus system uses the serial data line (SDA0) and serial clock line (SCL0) for data transfers. All the devices connected to the bus require open drain or open collector outputs which must be connected with a pull-up resistor.

Each of the devices connected to the bus has a unique address which can be set up using software. The devices always operate in a simple master/slave relationship, where the master functions as the master transmitter or master receiver. The I<sup>2</sup>C interface is a true multi-master bus with a collision detection function and arbitration function to prevent data from being lost if more than one master attempts to start data transfer at the same time.

### ■ I<sup>2</sup>C Protocol

Figure 23.7-1 shows the format required for data transfer.

Figure 23.7-1 Data Transfer Example



The slave address is transmitted after a start condition (S) is generated. This address is seven bits followed by the data direction bit ( $R/\bar{W}$ ) in the eighth bit position. Data is transmitted after the address. The data is eight bits followed by an acknowledgment.

Data can be transmitted continuously to the same slave address in consecutive units of eight bits plus acknowledgment.

Data transfer is always ended in the master stop condition (P). However, the repeated start condition (S) can be used to transmit the address which indicates a different slave without generating a stop condition.

## ■ Start Conditions

While the bus is idle (SCL0 and SDA0 are both at the logical "H" level), the master generates a start condition to start transmission. As shown in Figure 23.7-1, a start condition is triggered when the SDA0 line is changed from "H" to "L" while SCL0 = "H". This starts a new data transfer and commences master/slave operation.

A start condition can be generated in either of the following two ways.

- By writing "1" to the IBCR10:MSS bit while the I<sup>2</sup>C bus is not in use (IBCR10:MSS = 0, IBSR0:BB = 0, IBCR10:INT = 0, and IBCR00:ALF = 0). (Next, IBSR0:BB is set to "1" to indicate that the bus is busy.)
- By writing "1" to the IBCR10:SCC bit during an interrupt while in bus master mode (IBCR10:MSS = 1, IBSR0:BB = 1, IBCR10:INT = 1, and IBCR00:ALF = 0). (This generates a repeated start condition.)

Writing "1" to the IBCR10:MSS or IBCR10:SCC bit is ignored in other than the above cases. If another system is using the bus when "1" is written to the IBCR10:MSS bit, the IBCR00:ALF bit is set to "1".

## ■ Addressing

### ● Slave addressing in master mode

In master mode, IBSR0:BB and IBSR0:TRX are set to "1" after the start condition is generated, and the slave address in the IDDR0 register is output to the bus starting with the MSB. The address data consists of eight bits: the 7-bit slave address and the data transfer direction R/W bit (bit0 of IDDR0).

The acknowledgment from the slave is received after the address data is sent. SDA0 goes to "L" in the ninth clock cycle and the acknowledge bit from the receiving device is received (see Figure 23.7-1). In this case, the R/W bit (IDDR0:bit0) is inverted logically and stored in the IBSR0:TRX bit as "1" if the SDA level is "L".

### ● Addressing in slave mode

In slave mode, after the start condition is detected, IBSR0:BB is set to "1" and IBSR0:TRX is set to "0", and the data received from the master is stored in the IDDR0 register. After the address data is received, the IDDR0 and IAAR0 registers are compared. If the addresses match, IBSR0:AAS is set to "1" and an acknowledgment is sent to the master. Next, bit0 of the receive data (bit0 of the IDDR0 register) is saved in the IBSR0:TRX bit.

## ■ Data Transfer

If the MCU is addressed as a slave, data can be sent or received byte by byte with the direction determined by the R/W bit sent by the master.

Each byte to be output on the SDA0 line is fixed at eight bits. As shown in Figure 23.7-1, the receiver sends an acknowledgment to the sender by forcing the SDA0 line to the stable "L" level while the acknowledge clock pulse is "H". Data is transferred at one clock pulse per bit with MSB at the head. Sending and receiving an acknowledgment is required after each byte is transferred. Accordingly, nine clock pulses are required to transfer one complete data byte.



## ■ Acknowledgment

An acknowledgment is sent by the receiver in the ninth clock cycle for data byte transfer by the sender based on the following conditions.

An address acknowledgment is generated in the following cases.

- The received address matches the address set in IAAR0, and the address acknowledgment is output automatically (IBCR00:AACKX = 0).
- A general call address (00<sub>H</sub>) is received and the general call address acknowledgment output is enabled (IBCR10:GACKE = 1).

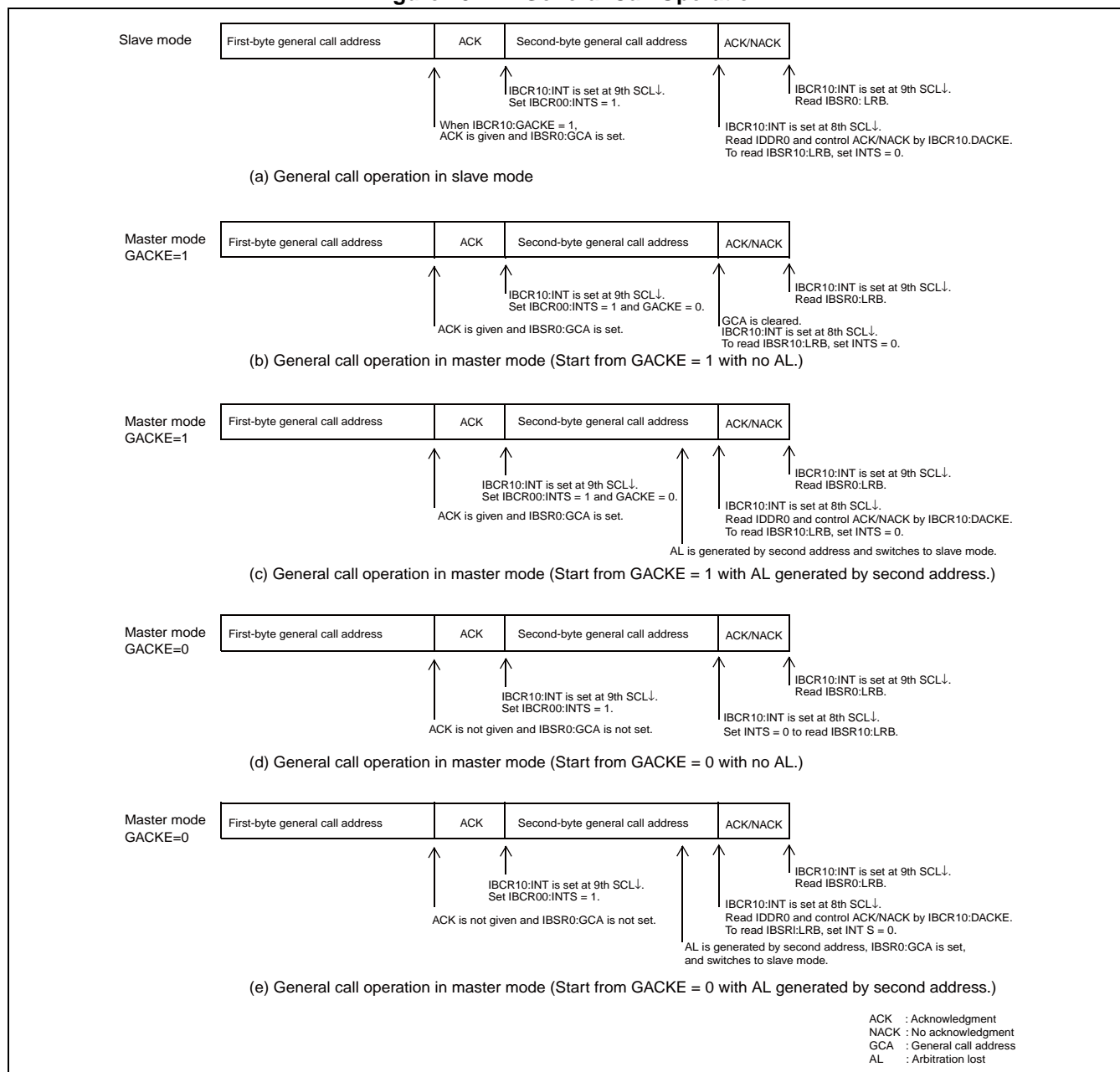
A data acknowledge bit used when data is received can be enabled or disabled by the IBCR10:DACKE bit. In master mode, a data acknowledgment is generated if IBCR10:DACKE = 1. In slave mode, a data acknowledgment is generated if an address acknowledgment has already been generated and IBCR10:DACKE = 1. The received acknowledgment is saved in IBSR0:LRB in the ninth SCL0 cycle.

- If the data ACK depends on the content of received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACKE) after writing "1" to the IBCR00:INTS bit (for example, by a previous transfer completion interrupt) so that the latest received data can be read.
- The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt triggered by the ninth SCL0 cycle). Accordingly, if ACK is read when the IBCR00:INTS bit is "1", you must write "0" to this bit in the transfer completion interrupt triggered by the eighth SCL0 cycle so that another transfer completion interrupt will be triggered by the ninth SCL0 cycle.

## General Call Address

A general call address consists of the start address byte (00<sub>H</sub>) and the second address byte that follows. To use a general call address, you must set IBCR10:GACKE=1 before the acknowledge of the first byte general call address. Also, the acknowledgment for the second address byte can be controlled as shown below.

**Figure 23.7-2 General Call Operation**



If this module sends a general call address at the same time as another device, you can determine whether the module successfully seized control of the bus by checking whether arbitration lost was detected when the second address byte was transferred. If arbitration lost was detected, the module goes to slave mode and continues to receive data from the master.

## ■ Stop Condition

The master can release the bus and end communications by generating a stop condition. Changing the SDA0 line from "L" to "H" while SCL0 is "H" generates a stop condition. This signals to the other devices on the bus that the master has finished communications (referred to below as "bus free"). However, the master can continue to generate start conditions without generating a stop condition. This is called a repeated start condition.

Writing "0" to the IBCR10:MSS bit during an interrupt while in bus master mode (IBCR10:MSS = 1, IBSR0:BB = 1, IBCR10:INT = 1, and IBCR00:ALF = 0) generates a stop condition and changes to slave mode. In other cases, writing "0" to the IBCR10:MSS bit is ignored.

## ■ Arbitration

The interface circuit is a true multi-master bus able to connect multiple master devices. Arbitration occurs when another master within the system simultaneously transfers data during a master transfer.

Arbitration occurs on the SDA0 line while the SCL0 line is at the "H" level. When the send data is "1" and the data on the SDA0 line is "L" at the master, this is treated as arbitration lost. In this case, data output is halted and IBCR00:ALF is set to "1". If this occurs, an interrupt is generated if arbitration lost interrupts have been enabled (IBCR00:ALE = 1). If IBCR00:ALF is set to "1", the module sets IBCR10:MSS = 0 and IBSR0:TRX = 0, clears TRX, and goes to slave receive mode.

If IBCR00:ALF is set to "1" when IBSR0:BB = 0, IBCR00:ALF is cleared only by writing "0". If IBCR00:ALF is set to "1" when IBSR0:BB = 1, IBCR00:ALF is cleared only by clearing IBCR10:INT to "0".

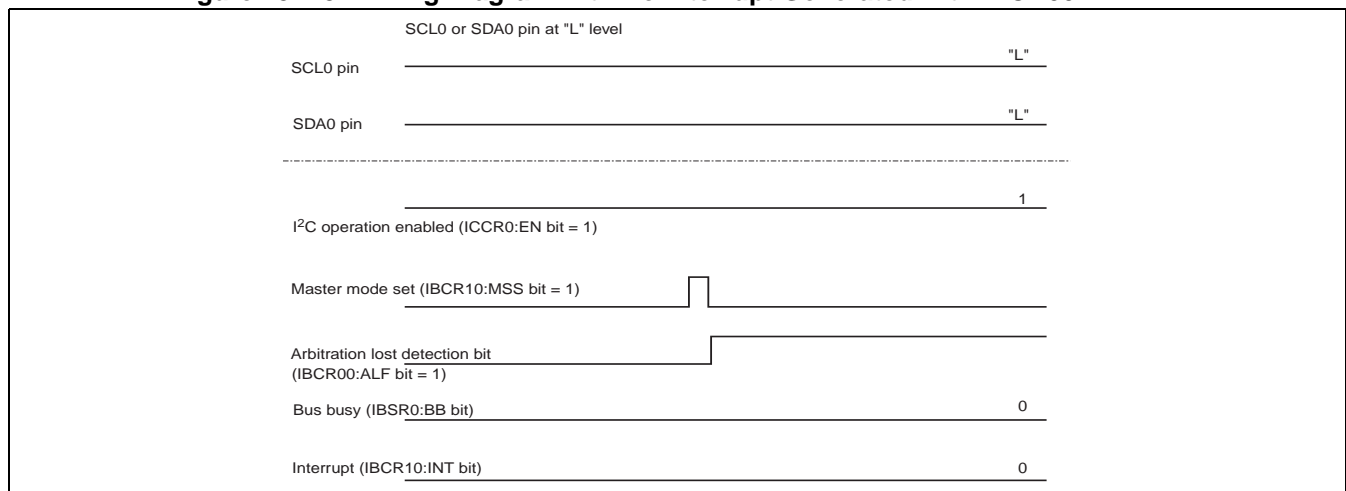
### ● Conditions for generating an arbitration lost interrupt when IBSR0:BB = 0

When a start condition is generated by the program (by setting the IBCR10:MSS bit to "1") at the timing shown in Figure 23.7-3 or Figure 23.7-4, interrupt generation (IBCR10:INT bit = 1) is prohibited by arbitration lost detection (IBCR00:ALF = 1).

- Conditions (1) in which no interrupt is generated due to arbitration lost

If the program triggers a start condition (by setting the IBCR10:MSS bit to "1") when no start condition has been detected (IBSR0:BB bit = 0) and the SDA0 and SCL0 line pins are at the "L" level.

**Figure 23.7-3 Timing Diagram with No Interrupt Generated with IBCR00:ALF = 1**

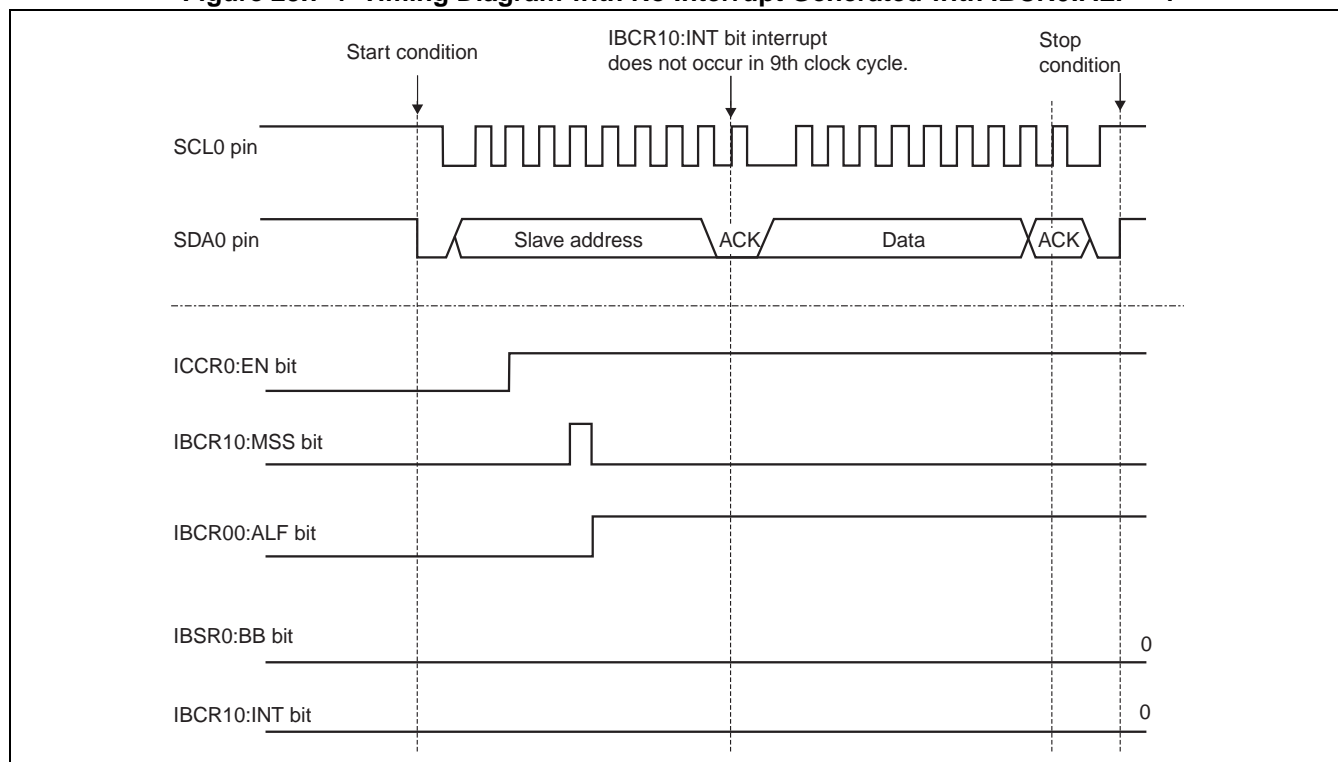


- Conditions (2) in which no interrupt is generated due to arbitration lost

If the program enables I<sup>2</sup>C operation (by setting the ICCR0:EN bit to "1") and triggers a start condition (by setting the IBCR10:MSS bit to "1") when the I<sup>2</sup>C bus is in use by another master.

This is because, as shown in Figure 23.7-4, this I<sup>2</sup>C module cannot detect the start condition (IBSR0:BB bit= 0) if another master starts communications on the I<sup>2</sup>C bus when the operation of this I<sup>2</sup>C module has been disabled (ICCR0:EN bit = 0).

**Figure 23.7-4 Timing Diagram with No Interrupt Generated with IBCR0:ALF = 1**



If this situation can occur, use the following procedure to set up the module from the software.

- 1) Trigger a start condition from the program (by setting the IBCR10:MSS bit to "1").
- 2) Check the IBCR00:ALF and IBSR0:BB bits in the arbitration lost interrupt.

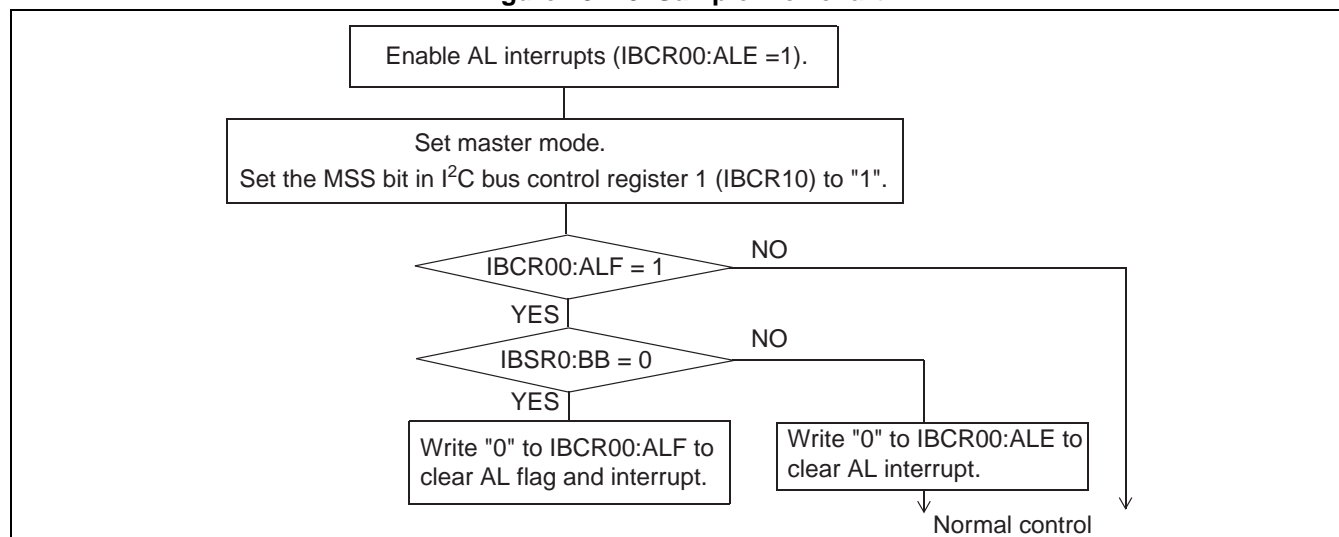
If IBCR00:ALF = 1 and IBSR0:BB = 0, clear the IBCR00:ALF bit to "0".

If IBCR00:ALF = 1 and IBSR0:BB = 1, clear the IBCR00:ALF bit to "0" and perform control as normal. (Normal control means writing "0" to the IBCR00:INT bit in the INT interrupt to clear IBCR00:ALF.)

In other cases, perform control as normal (Normal control means writing "0" to the IBCR00:INT bit in the INT interrupt to clear IBCR00:ALF.)

The following sample flowchart illustrates the procedure:

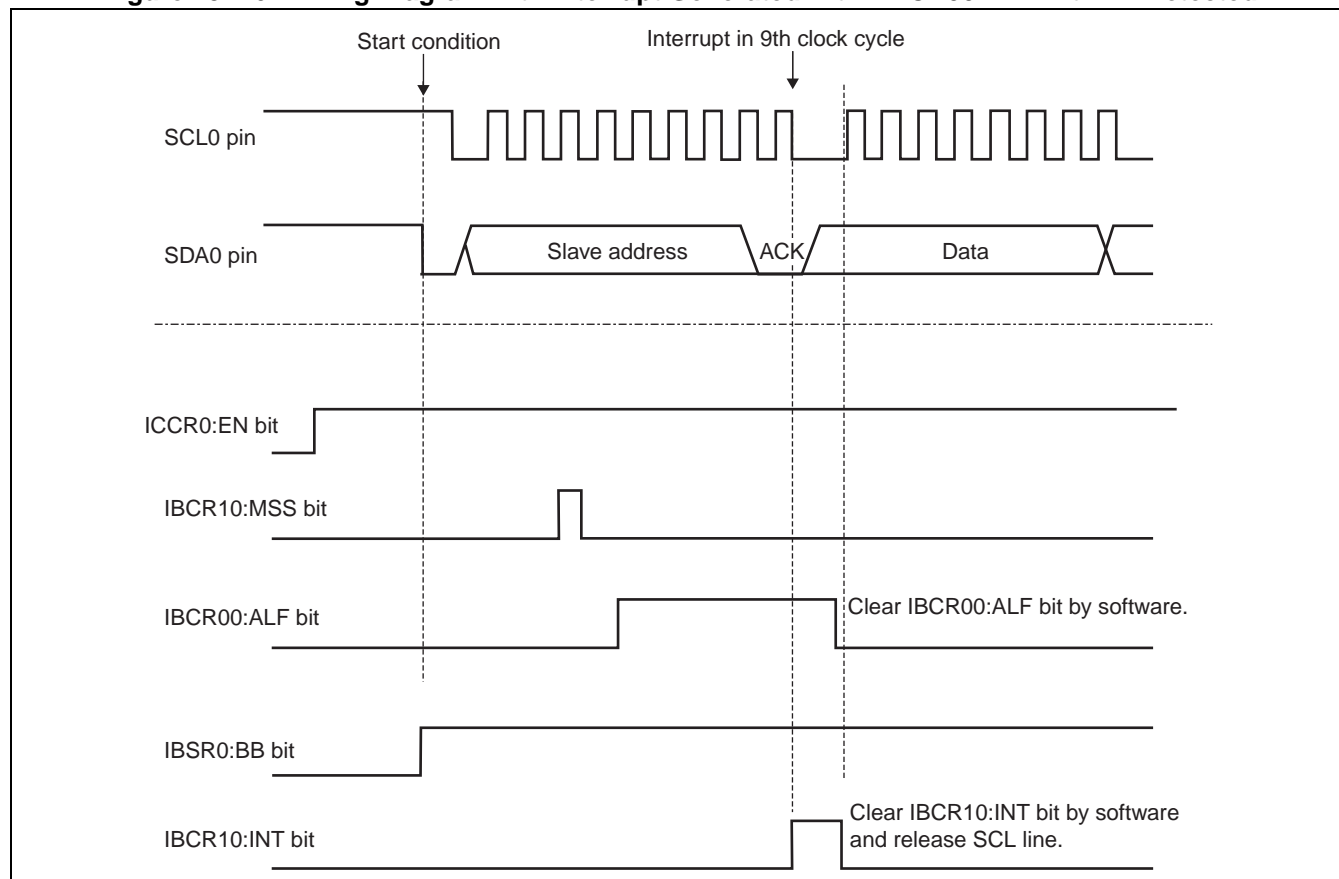
**Figure 23.7-5 Sample Flowchart**



● Example of generating an interrupt (IBCR10:INT bit = 1) with "IBCR00:ALF bit = 1" detected

If a start condition is generated by the program (by setting the IBCR10:MSS bit to "1") with the bus busy (IBSR0:BB bit = 1) and arbitration lost detected, a IBCR10:INT bit interrupt occurs upon detection of "IBCR00:ALF bit = 1".

**Figure 23.7-6 Timing Diagram with Interrupt Generated with "IBCR00:ALF Bit = 1" Detected**



## 23.7.2 Function to Wake up the MCU from Standby Mode

The wakeup function enables the I<sup>2</sup>C macro to be accessed while the MCU is in stop or watch mode.

### ■ Function to Wake Up the MCU from Standby Mode

The I<sup>2</sup>C macro includes a function to wake up the MCU from standby mode. The function is enabled by writing "1" to the IBCR00:WUE bit.

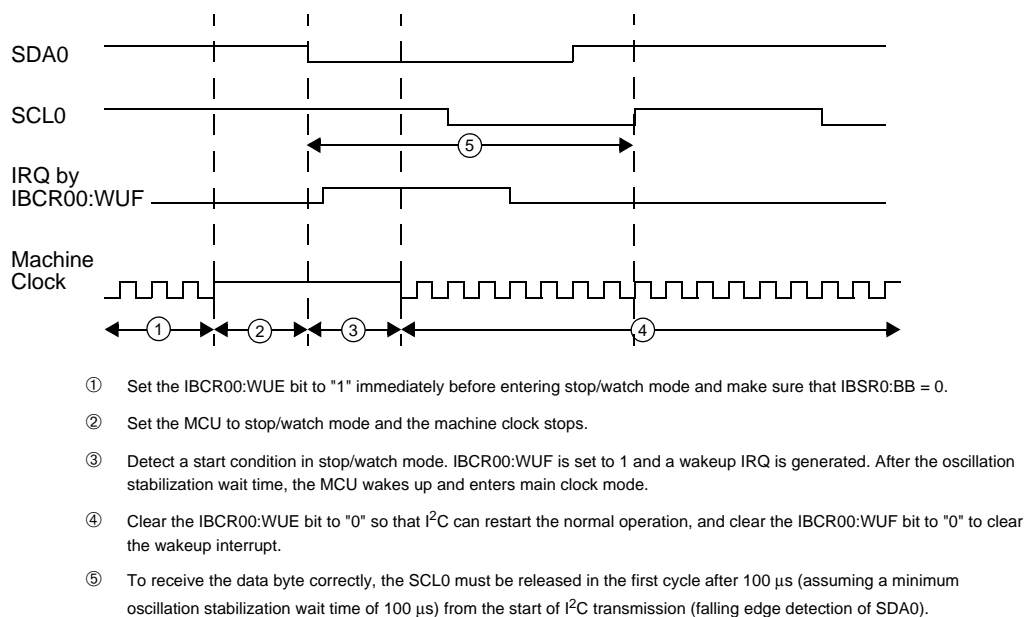
When the MCU is in stop/watch mode with the IBCR00:WUE bit containing "1", if a start condition is detected on the I<sup>2</sup>C bus, the wakeup interrupt request flag bit (IBCR00:WUF) is set to "1" and the wakeup interrupt request is generated to wake up the MCU from stop/watch mode.

- Set IBCR00:WUE to "1" immediately prior to setting the MCU to stop or watch mode. Similarly, clear IBCR00:WUE (by writing "0") after the MCU wakes up from stop or watch mode so that I<sup>2</sup>C operation can restart as soon as possible.
- The wakeup function only applies to the MCU stop and watch modes.

#### Note:

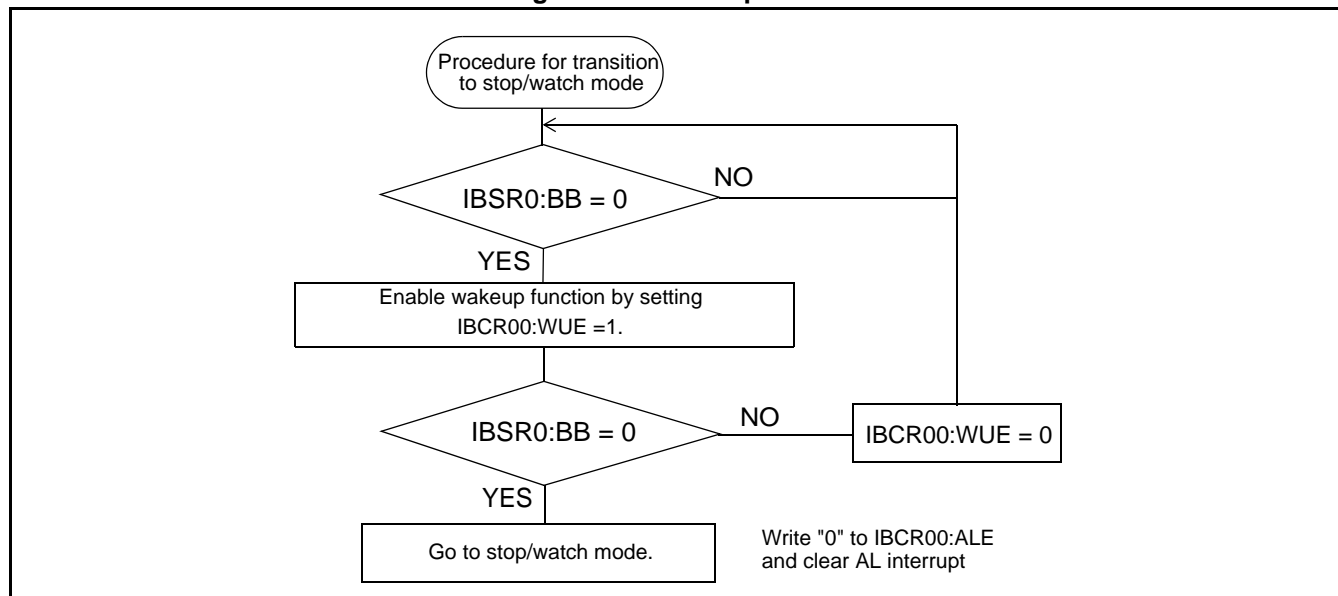
In PLL stop mode, a PLL oscillation stabilization wait time is required in addition to the oscillation stabilization wait time. This causes a very long delay between the MCU waking up and communications restarting.

**Figure 23.7-7 Comparison of Normal I<sup>2</sup>C Operation and Wakeup Operation**



The following sample flowchart illustrates the wakeup function.

**Figure 23.7-8 Sample Flow**



## 23.8 Notes on Use of I<sup>2</sup>C

---

This section summarizes notes on using the I<sup>2</sup>C interface.

---

### ■ Notes on Use of I<sup>2</sup>C

#### ● Notes on setting I<sup>2</sup>C interface registers

- Operation of the I<sup>2</sup>C interface must be enabled (ICCR0:EN) before setting the I<sup>2</sup>C bus control registers (IBCR00 and IBCR10).
- Setting the master/slave select bit (IBCR10:MSS) (by writing "1") starts data transfer.

#### ● Notes on setting the shift clock frequency

- The shift clock frequency can be calculated by determining the m, n, and DMBP values using the F<sub>sck</sub> equation in Table 23.5-4.
- "DMBP=1" may not be selected if the value of n is 4 (ICCR0:CS2 = CS1 = CS = 0).

#### ● Notes on priority for simultaneous writes

- Contention between next byte transfer and stop condition  
When "0" is written to IBCR10:MSS with IBCR10:INT cleared, the MSS bit takes priority and a stop condition develops.
- Contention between next byte transfer and start condition  
When "1" is written to IBCR10:SCC with IBCR10:INT cleared, the SCC bit takes priority and a start condition develops.

#### ● Notes on setup using software

- Do not select a repeated start condition (IBCR10:SCC=1) and slave mode (IBCR10:MSS=0) simultaneously.
- Execution cannot return from interrupt processing if the interrupt request enable bit is enabled (IBCR10:BEIE=1/IBCR10:INTE=1) with the interrupt request flag bit (IBCR10:BER/IBCR10:INT) containing "1". Be sure to clear the IBCR10:BER/IBCR10:INT bit.
- The following bits are cleared to "0" when I<sup>2</sup>C operation is disabled (ICCR0:EN=0):
  - AACKX, INTS, and WUE bits in the IBCR00 register
  - All the bits in the IBCR10 register except the BER and BEIE bits
  - All bits in the IBSR0 register

#### ● Notes on data acknowledgment

In slave mode, a data acknowledgment is generated in either of the following cases:

- When the received address matches the value in the address register (IAAR0) and IBCR00:AACKX = 0.
- When a general call address (00<sub>H</sub>) is received and IBCR10:GACKE = 1.



● Notes on selecting the transfer complete timing

- The transfer complete timing select bit (IBCR0:INTS) is valid only during data reception (IBSR0:TRX = 0 and IBSR0:FBT = 0).
- In cases other than data reception (IBSR0:TRX = 0 or IBSR0:FBT = 0), the transfer completion interrupt (IBCR10:INT) is always generated in the ninth SCL0 cycle.
- If the data ACK depends on the content of the received data (such as packet error checking used by the SM bus), control the data ACK by setting the data ACK enable bit (IBCR10:DACE) after writing "1" to the IBCR00:INTS bit (for example, using a previous transfer completion interrupt) to read latest received data.
- The latest data ACK (IBSR0:LRB) can be read after the ACK has been received (IBSR0:LRB must be read during the transfer completion interrupt in the ninth SCL0 cycle.) If ACK is read when the IBCR0:INTS bit is "1", therefore, you must write "0" to the IBCR00:INTS bit in the transfer completion interrupt in the eighth SCL0 cycle so that another transfer completion interrupt will occur in the ninth SCL0 cycle.

● Notes on using the MCU standby mode wakeup function

- Set IBCR00:WUE to "1" immediately prior to setting the MCU to stop or watch mode. Similarly, clear IBCR00:WUE (by writing "0") after the MCU wakes up from stop or watch mode so that I<sup>2</sup>C operation can restart as soon as possible.
- When a wakeup interrupt request occurs, the MCU wakes up after the oscillation stabilization wait time elapses. To prevent the data loss immediately after wakeup, design the system so that the SCL0 rises as the first cycle and the first bit must be transmitted as data after 100 μs (assuming a minimum oscillation stabilization wait time of 100 μs) from the wakeup due to start of I<sup>2</sup>C transmission (upon detection of the falling edge of SDA0).
- During a MCU standby mode, the status flags, state machine, and I<sup>2</sup>C bus outputs for the I<sup>2</sup>C function retain the states they had prior to entering the standby mode. To prevent a hang-up of the entire I<sup>2</sup>C bus system, make sure that IBSR0:BB = 0 before entering standby mode.
- The wakeup function does not support the transition of the MCU to stop or watch mode with IBSR0:BB = 1. If the MCU enters stop or watch mode with IBSR0:BB = 1, a bus error will occur upon detection of a start condition.
- In PLL stop mode, for example, the time from wakeup to the start of communication becomes longer than in stop/watch mode by the PLL oscillation stabilization wait time as the PLL oscillation stabilization wait time is required in addition to the oscillation stabilization wait time.
- To ensure correct operation of the I<sup>2</sup>C interface, always clear IBCR00:WUE to "0" after the MCU wakes up from stop or watch mode, regardless of whether this occurs due to the I<sup>2</sup>C wakeup function or the wakeup function for some other resource (such as an external interrupt).

## 23.9 Sample Programs for I<sup>2</sup>C

Fujitsu provides sample programs for operating the I<sup>2</sup>C interface.

### ■ I<sup>2</sup>C Sample Programs

For I<sup>2</sup>C sample programs, see "■ Sample Programs" in Preface.

### ■ Setting Methods Other than Those in Sample Programs

#### ● Enabling/disabling I<sup>2</sup>C operation

Use the I<sup>2</sup>C operation enable bit (ICCR0.EN).

Control	I <sup>2</sup> C operation enable bit (EN)
To disable I <sup>2</sup> C operation	Set the bit to "0".
To enable I <sup>2</sup> C operation	Set the bit to "1"

#### ● Selecting the I<sup>2</sup>C master or slave mode

Use the master/slave select bit (IBCR10.MSS).

Control	Master/slave select bit (MSS)
To select master mode	Set the bit to "1"
To select slave mode	Set the bit to "0".

#### ● Selecting the shift clock

Use the clock select bits (ICCR0. CS4/CS3/CS2/CS1/CS0).

#### ● Bypassing the divider-m when the shift clock frequency is generated

Use the divider-m bypass bit (ICCR0.DMBP).

Control	Divider-m bypass bit (DMBP)
To bypass divider m	Set the bit to "1"

● Controlling I<sup>2</sup>C address acknowledgment

Use the address acknowledge disable bit (IBCR00.AACKX).

Control	Address acknowledge disable bit (AACKX)
To enable address acknowledge output	Set the bit to "0".
To disable address acknowledge output	Set the bit to "1"

● Controlling I<sup>2</sup>C data acknowledgment

Use the data acknowledge enable bit (IBCR10.DACKE).

Control	Data acknowledge enable bit (DACKE)
To enable data acknowledge output	Set the bit to "1"
To disable data acknowledge output	Set the bit to "0".

● Controlling I<sup>2</sup>C general call address acknowledgment

Use the general call address acknowledge enable bit (IBCR10.GACKE).

Control	General call address acknowledge enable bit (GACKE)
To enable general call address acknowledge output	Set the bit to "1"
To disable general call address acknowledge output	Set the bit to "0".

● Restarting I<sup>2</sup>C communication

Use the start condition generation bit (IBCR10.SCC).

Control	Start condition generation bit (SCC)
To restart communication	Set the bit to "1"

● Selecting the I<sup>2</sup>C data reception transfer completion flag (INT)

Use the timing select bit (IBCR00.INTS) for the data reception transfer completion flag (INT).

Control	Timing select bit (INTS) for data reception transfer completion flag (INT)
To cause a transfer interrupt in the 9th SCL cycle	Set the bit to "0".
To cause a transfer interrupt in the 8th SCL cycle	Set the bit to "1"

## ● Interrupt related register

To set the interrupt level, use the following interrupt level setting register.

Channel	Interrupt level setting register	Interrupt vector
ch.0	Interrupt level register (ILR2) Address: 0007B <sub>H</sub>	#10 Address: 0FFE6 <sub>H</sub>

## ● Enabling, disabling, and clearing interrupts

### • Transfer interrupt

(Data transfer completion interrupt)

To enable interrupts, use the interrupt request enable bit (IBCR10.INTE).

Control	Interrupt request enable bit (INTE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1"

To clear interrupt requests, use the interrupt request flag (IBCR10.INT).

Control	Interrupt request flag (INT)
To clear interrupt requests	Set the bit to "0".

(Bus error generation interrupt)

To enable interrupts, use the interrupt request enable bit (IBCR10.BEIE).

Control	Interrupt request enable bit (BEIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1"

To clear interrupt requests, use the interrupt request flag (IBCR10.BER).

Control	Interrupt request flag (BER)
To clear interrupt requests	Set the bit to "0".

- Stop interrupt  
(Stop condition detection interrupt)  
To enable interrupts, use the interrupt request enable bit (IBCR00.SPE).

Control	Interrupt request enable bit (SPE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupt requests, use the interrupt request flag (IBCR00.SPF).

Control	Interrupt request flag (SPF)
To clear interrupt requests	Set the bit to "0".

- (Arbitration lost detection interrupt)  
To enable interrupts, use the interrupt request enable bit (IBCR00.ALE).

Control	Interrupt request enable bit (ALE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupt requests, use the interrupt request flag (IBCR00.ALF).

Control	Interrupt request flag (ALF)
To clear interrupt requests	Set the bit to "0".

- (Start condition detection interrupt)  
To enable interrupts, use the interrupt request enable bit (IBCR00.WUE).

Control	Interrupt request enable bit (WUE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupt requests, use the interrupt request flag (IBCR00.WUF).

Control	Interrupt request flag (WUF)
To clear interrupt requests	Set the bit to "0".

# **CHAPTER 24**

---

## **8/10-BIT A/D CONVERTER**

**This chapter describes the functions and operations of the 8/10-bit A/D converter.**

- 24.1 Overview of 8/10-bit A/D Converter
- 24.2 Configuration of 8/10-bit A/D Converter
- 24.3 Pins of 8/10-bit A/D Converter
- 24.4 Registers of 8/10-bit A/D Converter
- 24.5 Interrupts of 8/10-bit A/D Converter
- 24.6 Operations of 8/10-bit A/D Converter and Its Setup  
Procedure Examples
- 24.7 Notes on Use of 8/10-bit A/D Converter
- 24.8 Sample Programs for 8/10-bit A/D Converter

## **24.1 Overview of 8/10-bit A/D Converter**

---

**The 8/10-bit A/D converter is a 10-bit successive approximation type of 8/10-bit A/D converter. It can be started via software, external trigger, and internal clock, with one input signal selected from among multiple analog input pins.**

---

### **■ A/D Conversion Functions**

The A/D converter converts analog voltages (input voltages) input to an analog input pin to 10-bit digital values.

- One of multiple analog input pins can be selected.
- The conversion speed is programmable to be configured (selected according to the operating voltage and frequency).
- An interrupt is generated when A/D conversion completes.
- The completion of conversion can also be checked with the ADI bit in the ADC1 register.

To activate A/D conversion functions, follow one of the methods given below.

- Activation using the AD bit in the ADC1 register
- Continuous activation using the external pin (ADTG)
- Continuous activation using the 8/16-bit compound timer output TO00

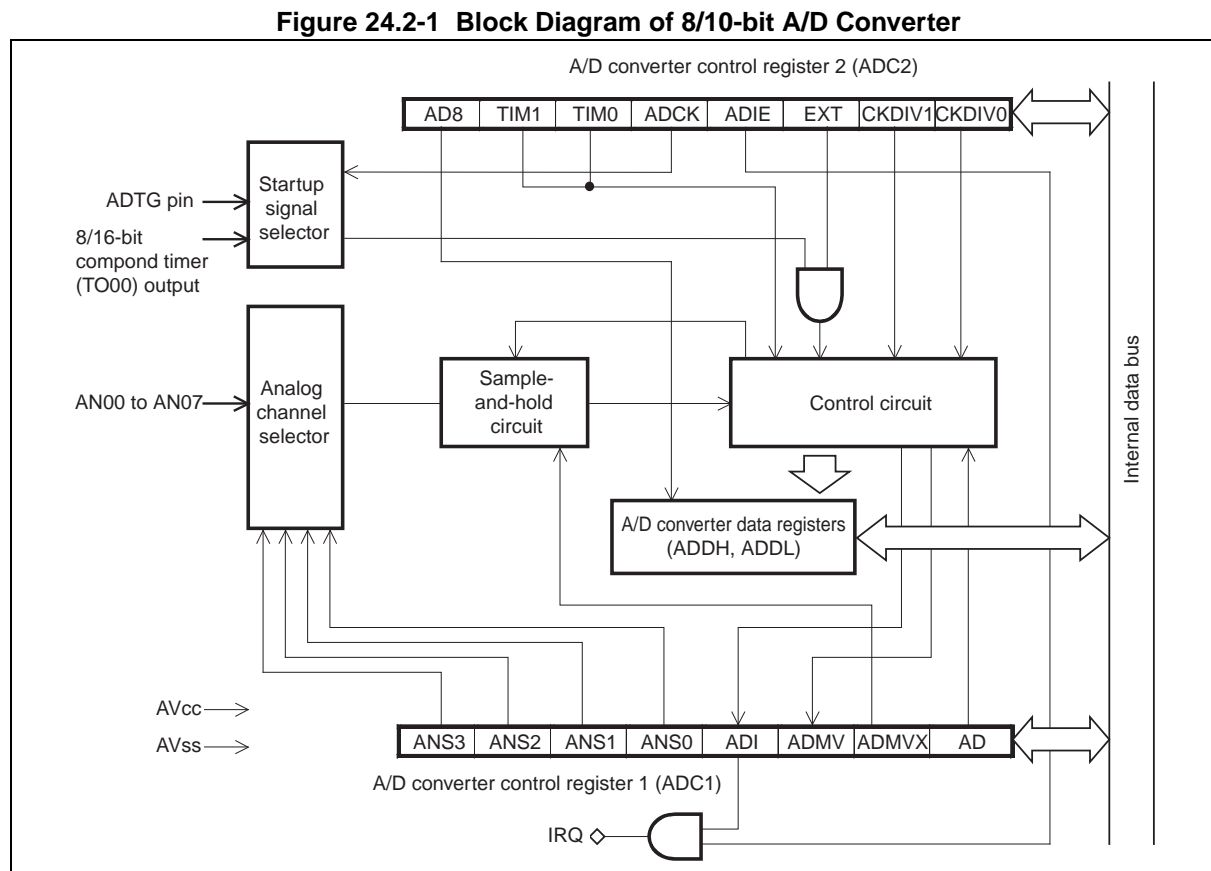
## 24.2 Configuration of 8/10-bit A/D Converter

The 8/10-bit A/D converter consists of the following blocks:

- Clock selector (input clock selector for starting A/D conversion)
- Analog channel selector
- Sample-and-hold circuit
- Control circuit
- A/D converter data registers (ADDH, ADDL)
- A/D converter control register 1 (ADC1)
- A/D converter control register 2 (ADC2)

### ■ Block Diagram of 8/10-bit A/D Converter

Figure 24.2-1 shows a block diagram of the 8/10-bit A/D converter.





● Clock selector

This block selects the A/D conversion clock with continuous activation enabled (ADC2:EXT = 1).

● Analog channel selector

This circuit selects one of multiple analog input pins.

● Sample-and-hold circuit

This circuit holds the input voltage selected by the analog channel selector. This enables A/D conversion to be performed without being affected by variation in input voltage during conversion (comparison) by sampling and holding the input voltage immediately after starting A/D conversion.

● Control circuit

The A/D conversion function determines the values in the 10-bit A/D converter data register sequentially from MSB to LSB based on the signals from the comparator. When conversion is completed, the A/D conversion function sets the interrupt request flag bit (ADC1: ADI).

● A/D converter data registers (ADDH/ADDL)

The high-order two bits of 10-bit A/D data are stored in the ADDH register; the low-order eight bits are stored in the ADDL register.

Setting the A/D conversion precision bit (ADC2:AD8) to "1" provides 8-bit precision, storing the upper eight bits of the 10-bit A/D data in the ADDL register.

● A/D converter control register 1 (ADC1)

This register is used to enable and disable functions, select an analog input pin, check statuses, and control interrupts.

● A/D converter control register 2 (ADC2)

This register is used to select an input clock, enable and disable interrupts, and select functions.

■ Input Clock

The 8/10-bit A/D converter uses the output clock from the prescaler as the input clock (operation clock).

## 24.3 Pins of 8/10-bit A/D Converter

---

This section describes the pins of the 8/10-bit A/D converter.

---

### ■ Pins of 8/10-bit A/D Converter

MB95110B/M series has 8 channels of analog input pin.

Analog input pins also serve as general-purpose I/O ports.

#### ● AN07 to AN00 Pin

AN07 to AN00: When using the A/D conversion function, input the analog voltage you wish to convert to one of these pins. Each of the pins serves as an analog input pin by selecting it using the analog input channel select bits (ADC1: ANS0 to ANS3) with the corresponding bit in the port direction register (DDR) set to 0. Even when the 8/10-bit A/D converter is used, the pins not used for analog input can be used as general-purpose I/O ports.

Note that the number of analog input pins differs depending on the series.

#### ● ADTG Pin

ADTG : This is a pin used to activate A/D conversion function by external trigger.

#### ● AV<sub>CC</sub> pin

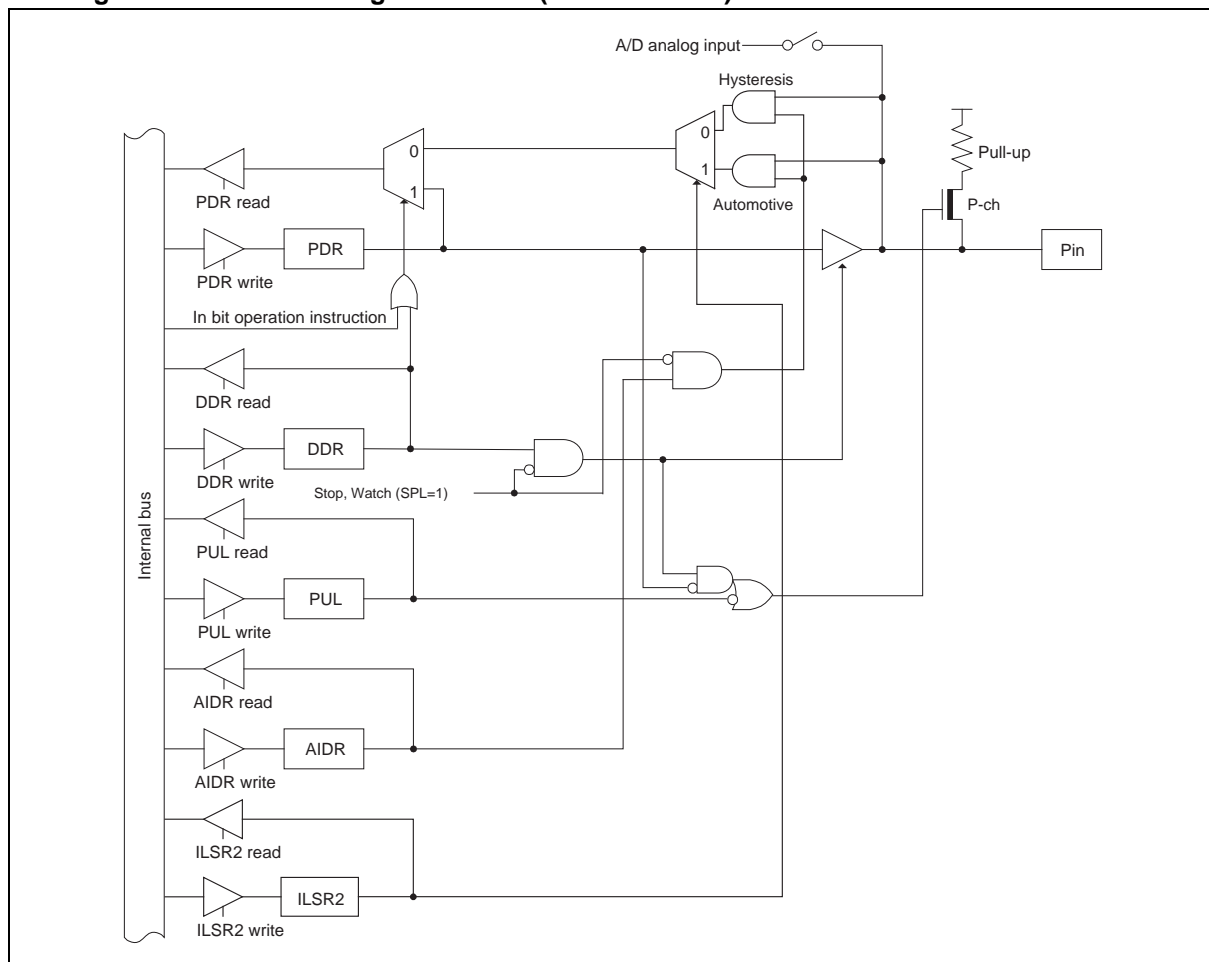
AV<sub>CC</sub>: This is a 8/10-bit A/D converter power supply pin. Use this at the same potential as V<sub>CC</sub>. If A/D conversion precision is demanded, you should take measures to ensure that V<sub>CC</sub> noise does not enter AV<sub>CC</sub>, or use a separate power source. You should connect this pin to a power source even when the 8/10-bit A/D converter is not being used.

#### ● AV<sub>SS</sub> pin

AV<sub>SS</sub>: This is a ground pin of the 8/10-bit A/D converter. Use this at the same potential as V<sub>SS</sub>. When A/D conversion precision is required, take measures to ensure that the V<sub>SS</sub> noise does not interfere with AV<sub>SS</sub>. You should connect this pin to a ground (GND) even when the 8/10-bit A/D converter is not being used.

■ **Block Diagram of Pins Related to 8/10-bit A/D Converter**

**Figure 24.3-1 Block Diagram of Pins (AN00 to AN07) Related to 8/10-bit A/D Converter**



## MB95110B/M Series

### 24.4 Registers of 8/10-bit A/D Converter

The 8/10-bit A/D converter has four registers: A/D converter control register 1 (ADC1), A/D converter control register 2 (ADC2), A/D converter data register upper (ADDH), and A/D converter data register lower (ADDL).

#### ■ List of 8/10-bit A/D Converter Registers

Figure 24.4-1 lists the registers of the 8/10-bit A/D converter.

**Figure 24.4-1 Registers of 8/10-bit A/D Converter**

8/10-bit A/D converter control register 1 (ADC1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006C <sub>H</sub>	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R(RM1),W	R/WX	R/W	R0,W	

8/10-bit A/D converter control register 2 (ADC2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006D <sub>H</sub>	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

8/10-bit A/D converter data register upper (ADDH)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006E <sub>H</sub>	—	—	—	—	—	—	SAR9	SAR8	00000000 <sub>B</sub>
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	

8/10-bit A/D converter data register lower (ADDL)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
006F <sub>H</sub>	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1	SAR0	00000000 <sub>B</sub>
	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/W : Readable/writable (Read value is the same as write value)

R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)

R/WX : Read only (Readable, writing has no effect on operation)

R0, W : Write only (Writable, "0" is read)

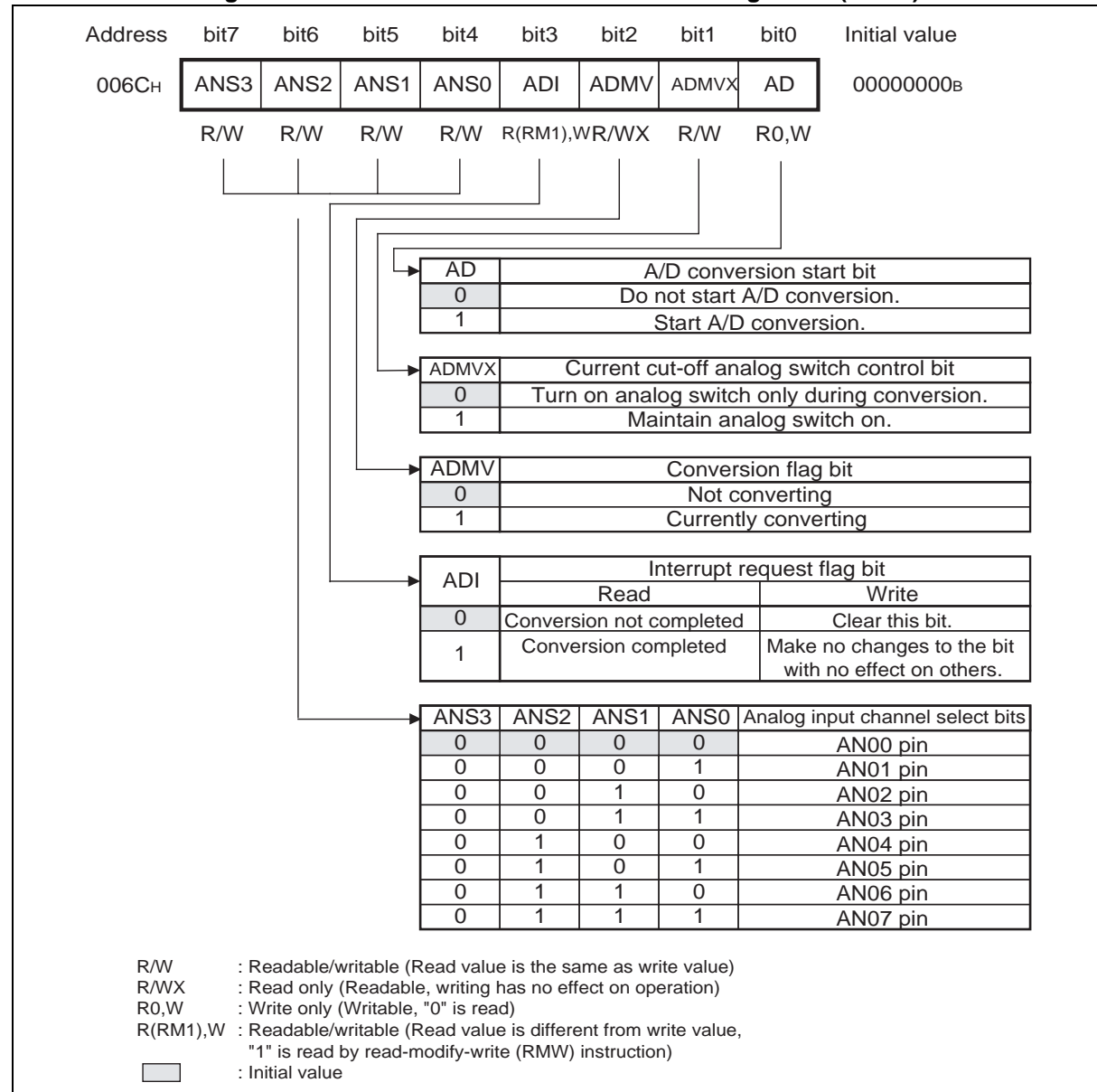
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

## 24.4.1 8/10-bit A/D Converter Control Register 1 (ADC1)

8/10-bit A/D converter control register 1 (ADC1) is used to enable and disable individual functions of the 8/10-bit A/D converter, select an analog input pin, and to check the states.

### ■ 8/10-bit A/D Converter Control Register 1 (ADC1)

Figure 24.4-2 8/10-bit A/D Converter Control Register 1 (ADC1)



Do not select the unusable channel for this series by analog input channel select bits (ANS3 to ANS0).

Table 24.4-1 Functions of Bits in 8/10-bit A/D Converter Control Register 1 (ADC1)

Bit name		Function
bit7, bit6, bit5, bit4	ANS3, ANS2, ANS1, ANS0: Analog input channel select bits	Select the analog input pin to be used from among AN00 to AN07. Note that the number of analog input pins differs depending on the series. When A/D conversion is activated (AD = 1) via software (ADC2: EXT = 0), these bits can be updated at the same time. Note: When the ADMV bit is "1", do not update these bits. The pins not used as analog input pins can be used as general-purpose ports.
bit3	ADI: Interrupt request flag bit	Detects the termination of A/D conversion. <ul style="list-style-type: none"> <li>When the A/D conversion function is used, the bit is set "1" upon termination of A/D conversion.</li> <li>Interrupt requests are output when this bit and the interrupt request enable bit (ADC2: ADIE) are both "1".</li> <li>When written to this bit, "0" clears it; "1" leaves it unchanged with no affect on others.</li> <li>When read by a read-modify-write (RMW) instruction, the bit returns "1".</li> </ul>
bit2	ADMV: Conversion flag bit	Indicates that conversion is ongoing during execution of the A/D conversion function. The bit contains "1" during conversion. This bit is read-only. Any value attempted to be written is meaningless and has no effect on operation.
bit1	ADMVX: Analog switch for shutting down control bit	Controls the analog switch for shutting down the internal reference power supply. When the external impedance of the AVR pin is high, rush current flows immediately after A/D startup and may affect A/D conversion precision. In this kind of situation, this can be avoided by setting this bit to "1" before A/D startup. Set the bit to "0" before switching to standby mode, in order to reduce current consumption. Note that some series do not have AVR pins, and are internally connected to AV <sub>CC</sub> .
bit0	AD: A/D conversion startup bit	Starts the A/D conversion function via software. Writing "1" to the bit starts the A/D conversion function. Note: Writing "0" to this bit will not stop operation of the A/D conversion function. The value read is always "0". A/D conversion startup by this bit is disabled with EXT=1. A/D converter re-starts by writing "1" to this bit during A/D conversion with EXT = 0.

## 24.4.2 8/10-bit A/D Converter Control Register 2 (ADC2)

8/10-bit A/D converter control register 2 (ADC2) selects the 8/10-bit A/D converter function, selects the input clock, and performs interrupt and status checking.

### ■ 8/10-bit A/D Converter Control Register 2 (ADC2)

Figure 24.4-3 8/10-bit A/D Converter Control Register 2 (ADC2)

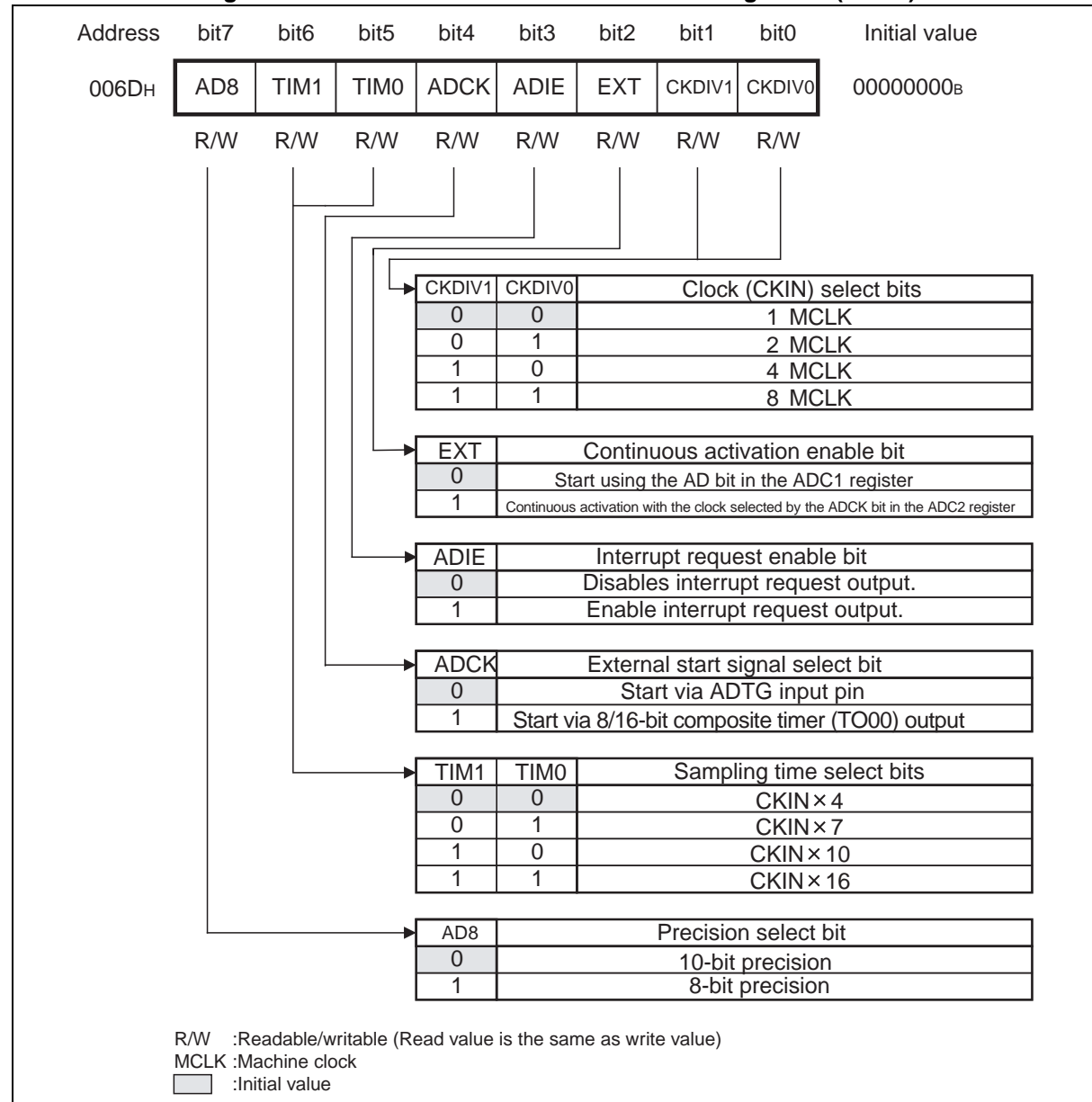


Table 24.4.-2 Functions of Bits in 8/10-bit A/D Converter Control Register 2 (ADC2)

Bit name		Function
bit7	AD8: Precision select bit	<p>This bit selects the resolution of A/D conversion.</p> <p><b>When set to "0":</b> The bit selects 10-bit precision.</p> <p><b>When set to "1":</b> The bit selects 8-bit precision, in which case eight bits of data can be read from the ADDL register.</p> <p>Note: The data bits used are different depending on the resolution. Update this bit only with A/D operation stopped before starting conversion.</p>
bit6, bit5	TIM1, TIM0: Sampling time select bits	<p>Set the sampling time.</p> <ul style="list-style-type: none"> <li>Change this sampling time setting depending on the operating conditions (voltage and frequency).</li> <li>The CKIN value is determined by the clock select bits (ADC2:CKDIV1, DKDIV0).</li> </ul> <p>Note: Update this bit only with A/D operation stopped.</p>
bit4	ADCK: External start signal select bit	Selects the start signal for external start (ADC2:EXT = 1).
bit3	ADIE: Interrupt request enable bit	<p>Enables or disables output of interrupts to the interrupt controller.</p> <p>Interrupt requests are output with both of this bit and the interrupt request flag bit (ADC1: ADI) set to "1".</p>
bit2	EXT: Continuous activation enable bit	Selects whether to activate the A/D conversion function via software, or continuously upon detection of the rise of the input clock signal.
bit1, bit0	CKDIV1, CKDIV0: Clock select bits	<p>Select the clock to use for A/D conversion. The input clock is generated by the prescaler. See "CHAPTER 6 CLOCK CONTROLLER".</p> <ul style="list-style-type: none"> <li>The sampling time can also be changed via this clock selection.</li> <li>Change this setting depending on the operating conditions (voltage and frequency).</li> </ul> <p>Note: Update this bit only with A/D operation stopped.</p>



### 24.4.3 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)

The 8/10-bit A/D converter data registers upper/lower (ADDH, ADDL) contain the results of 10-bit A/D conversion.

The high-order two bits of 10-bit data correspond to the ADDH register; the low-order eight bits correspond to the ADDL register.

#### ■ 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)

**Figure 24.4-4 8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL)**

ADDH	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	—	—	—	—	—	—	SAR9	SAR8	00000000 <sub>B</sub>
006E <sub>H</sub>	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/WX	R/WX	
ADDL	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
Address	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	SAR1	SAR0	00000000 <sub>B</sub>
006F <sub>H</sub>	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	

R/WX : Read only (Readable, writing has no effect on operation)  
R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)

The upper two bits of 10-bit A/D data correspond to bits1 and 0 in the ADDH register; the lower eight bits correspond to bit15 to bit8 in the ADDL register.

Set the AD8 bit in the ADC2 register to "1" to select 8-bit precision mode, so that 8-bit data can be read from the ADDL register.

These registers are read-only. Writing has no effect on the operation.

During 8-bit conversion, SAR8 and SAR9 hold "0".

#### ● A/D Conversion Functions

When A/D conversion is started, the results of conversion are finalized and stored in these registers after the conversion time according to the register settings has passed. After A/D conversion finishes, therefore, read the A/D data registers (conversion results), write "0" to the ADI bit (bit3) in the ADC1 register before the next A/D conversion terminates, then after A/D conversion finishes, clear the flag. During A/D conversion, the registers contain the values resulting from the last conversion performed.

## 24.5 Interrupts of 8/10-bit A/D Converter

---

**An interrupt source of the 8/10-bit A/D converter is Completion of conversion when A/D conversion functions are operating.**

---

### ■ Interrupts During 8/10-bit A/D Converter Operation

When A/D conversion is completed, the interrupt request flag bit (ADC1: ADI) is set to "1". Then if the interrupt request enable bit is enabled (ADC2: ADIE = 1), an interrupt request is issued to the interrupt controller. Write "0" to the ADI bit using the interrupt service routine to clear the interrupt request.

The ADI bit is set when A/D conversion is completed, irrespective of the value of the ADIE bit.

The CPU cannot return from interrupt processing if the interrupt request flag bit (ADC1: ADI) is 1 with interrupt requests enabled (ADC2: ADIE = 1). Be sure to clear the ADI bit within the interrupt service routine.

### ■ Register and Vector Table Related to 8/10-bit A/D Converter Interrupts

**Table 24.5-1 Register and Vector Table Related to 8/10-bit A/D Converter Interrupts**

Interrupt source	Interrupt request number	Interrupt level setting register		Vector table address	
		Registers	Setting bit	Upper	Lower
8/10-bit A/D	IRQ18	ILR4	L18	FFD6 <sub>H</sub>	FFD7 <sub>H</sub>

Refer to "APPENDIX B Table of Interrupt Causes" for the interrupt request numbers and vector tables of all peripheral functions.

## 24.6 Operations of 8/10-bit A/D Converter and Its Setup Procedure Examples

The EXT bit in the ADC1 register can be used to select the software activation or continuous activation of the 8/10-bit A/D converter.

### ■ Operations of 8/10-bit A/D Converter's Conversion Function

#### ● Software activation

The settings shown in Figure 24.6-1 are required for software activation of the A/D conversion function.

**Figure 24.6-1 Settings for A/D Conversion Function (Software Activation)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD
	⊙	⊙	⊙	⊙	⊙	⊙	⊙	1
ADC2	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0
	⊙	⊙	⊙	X	⊙	0	⊙	⊙
ADDH	—	—	—	—	—	—	A/D converted value retained	
ADDL	A/D converted value is retained.							

⊙: Used bit  
x: Unused bit  
0 : Set to "0"  
1 : Set to "1"

When A/D conversion is activated, the A/D conversion function starts working. In addition, even during conversion, the A/D conversion function can be reactivated.

● Continuous activation

The settings shown in Figure 24.6-2 are required for continuous activation of the A/D conversion function.

**Figure 24.6-2 Settings for A/D Conversion Function (Continuous Activation)**

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADC1	ANS3	ANS2	ANS1	ANS0	ADI	ADMV	ADMVX	AD
	⊙	⊙	⊙	⊙	⊙	⊙	⊙	X
ADC2	AD8	TIM1	TIM0	ADCK	ADIE	EXT	CKDIV1	CKDIV0
	⊙	⊙	⊙	⊙	⊙	1	⊙	⊙
ADDH	—	—	—	—	—	—	A/D converted value retained	

⊙: Used bit  
x: Unused bit  
1 : Set to "1"

When continuous activation is enabled, A/D conversion is activated at the rising edge of the selected input clock to start the A/D conversion function. Continuous activation is stopped by disabling it (ADC2:EXT = 0).

## ■ Operations of A/D Conversion Function

This section details the operations of the 8/10-bit A/D converter.

- 1) When A/D conversion is started, the conversion flag bit is set (ADC1:ADMV = 1) and the selected analog input pin is connected to the sample-and-hold circuit.
- 2) The voltage at the analog input pin is loaded into the sample-and-hold capacitor in the sample-and-hold circuit during the sampling cycle. This voltage is held until A/D conversion has been completed.
- 3) The comparator in the control circuit compares the voltage loaded into the sample-and-hold capacitor with the A/D conversion reference voltage, from the most significant bit (MSB) to the least significant bit (LSB), and then sends the results to the ADDH and ADDL registers. After the results have been completely transferred, the conversion flag bit is cleared (ADC1:ADMV = 0) and the interrupt request flag bit is set (ADC1:ADI = 1).

### Notes:

- When the A/D conversion function is used, the contents of the ADDH and ADDL registers are retained upon completion of A/D conversion. During A/D conversion, the values resulting from the last conversion are loaded.
- Do not re-select the analog input channel (ADC1: ANS3 to ANS0) while the A/D conversion function is running, in particular, during continuous activation. Disable continuous activation (ADC2: EXT = 0) before re-selecting the analog input channel.
- Starting the reset, stop, or watch mode stops the 8/10-bit A/D converter and initializes each register.

### ■ Setup Procedure Example

Follow the procedure below to set up the 8/10-bit A/D converter:

#### ● Initial setting

- 1) Set the port for input (DDR3).
- 2) Set the interrupt level (ILR4).
- 3) Enable A/D input (ADC1:ANS0 to ANS3).
- 4) Set the sampling time (ADC2:TIM1, TIM0).
- 5) Select the clock (ADC2:CKDIV1, CKDIV0).
- 6) Set A/D conversion properties (ADC2:AD8).
- 7) Select the operation mode (ADC2:EXT).
- 8) Select the startup trigger (ADC2:ADCK).
- 9) Enable interrupts (ADC2:ADIE=1).
- 10) Activate A/D (ADC1:AD=1).

#### ● Interrupt processing

- 1) Clear the interrupt request flag (ADC1:ADI=0).
- 2) Read converted values (ADDH, ADDL).
- 3) Activate A/D (ADC1:AD=1).

## 24.7 Notes on Use of 8/10-bit A/D Converter

---

This section summarizes notes on using the 8/10-bit A/D converter.

---

### ■ Notes on Use of 8/10-bit A/D Converter

#### ● Notes on programmed setup

- When the A/D conversion function is used, the contents of the ADDH and ADDL registers are retained upon completion of A/D conversion. During A/D conversion, the values resulting from the last conversion are loaded.
- Do not re-select the analog input channel (ADC1: ANS3 to ANS0) while the A/D conversion function is running, in particular, during continuous activation. Disable continuous activation (ADC2: EXT = 0) before re-selecting the analog input channel.
- Starting the reset, stop, or watch mode stops the 8/10-bit A/D converter and initializes each register.
- The CPU cannot return from interrupt processing if the interrupt request flag bit (ADC1: ADI) is 1 with interrupt requests enabled (ADC2: ADIE = 1). Be sure to clear the ADI bit within the interrupt processing routine.

#### ● Note on interrupt requests

If A/D conversion is reactivated (ADC1: AD = 1) and terminated at the same time, the interrupt request flag bit (ADC1: ADI) is set.

#### ● Error

As  $|AVR - AV_{SS}|$  decreases, an error increases relatively.

#### ● 8/10-bit A/D converter and analog input power-on/shut-down sequences

Turn on the 8/10-bit A/D converter power supply ( $AV_{CC}$ ,  $AV_{SS}$ ) and analog input (AN00 to AN07) at the same as or after turning on the digital power supply ( $V_{CC}$ ).

In addition, turn off the digital power supply ( $V_{CC}$ ) either at the same time as or after turning off the 8/10-bit A/D converter power supply ( $AV_{CC}$ ,  $AV_{SS}$ ) and analog input (AN00 to AN07).

Be careful not to let the  $AV_{CC}$ ,  $AV_{SS}$ , and analog input exceed the voltage of the digital power supply when turning the 8/10-bit A/D converter on and off.

#### ● Conversion time

The conversion speed of the A/D conversion function is affected by the clock mode, main clock oscillation frequency, and main clock speed switching (gear function).

Example: Sampling time =  $CKIN \times (\text{ADC2: TIM1/TIM0 setting})$

Compare time =  $CKIN \times 10$  (fixed value) + MCLK

AD start processing time: Min. = MCLK + MCLK

Max. = MCLK + CKIN

Conversion time = A/D start processing time + sampling time + compare time

- The error max. 1 CKIN-1MCLK may occur depending on the timing of AD startup.
- Program the software satisfied with "sampling time" and "compare time" in A/D converter of data sheet.

## 24.8 Sample Programs for 8/10-bit A/D Converter

Fujitsu provides sample programs to operate the 8/10-bit A/D converter.

### ■ Sample Programs for 8/10-bit A/D Converter

For sample programs for the 8/10-bit A/D converter, see "■ Sample Programs" in "Preface".

### ■ Setting Methods not Covered by Sample Programs

- Selecting the operating clock for the 8/10-bit A/D converter

Use the clock select bits (ADC2.CKDIV1/CKDIV0) to select the operating clock.

- Selecting the sampling time of the 8/10-bit A/D converter

Use the sampling time select bits (ADC2.TIM1/TIM0) to select the sampling time.

- Controlling the analog switch for internal reference power shutdown of the 8/10-bit A/D converter

Use the analog switch control bit (ADC1.ADMVX) to control the internal reference power shutdown analog switch.

Control item	Analog switch control bit (ADMVX)
To turn off internal reference power supply	Set the bit to "0".
To turn on internal reference power supply	Set the bit to "1".

- Selecting the 8/10-bit A/D converter activation method

Use the continuous activation enable bit (ADC2.EXT) to select the startup trigger.

A/D startup factor	Continuous activation enable bit (EXT)
To select the software trigger	Set the bit to "0".
To select the input clock rising signal	Set the bit to "1".

- Generating a software trigger

Use the A/D conversion start bit (ADC1.AD) to generate a software trigger.

Operation	A/D conversion start bit (AD)
To generate a software trigger	Set the bit to "1".

- Activation using the input clock

A startup trigger is generated at the rise of the input clock signal.

To select the input clock, use the external start signal select bit (ADC2.ADCK).

Input clock	External start signal select bit (ADCK)
To select the ADTG input pin	Set the bit to "0".
To select the 8/16-bit compound timer (TO00)	Set the bit to "1".

● Selecting the A/D conversion precision

To select the precision of conversion results, use the precision select bit (ADC2.AD8).

Operation mode	Precision select bit (AD8)
To select 10-bit precision	Set the bit to "0".
To select 8-bit precision	Set the bit to "1".

● Using analog input pins

To select an analog input pin, use the analog input channel select bits (ADC1.ANS3 to ANS0).

Operation	Analog input channel select bits (ANS3 to ANS0)
To use the AN0 pin	Set the pins to "0000 <sub>B</sub> ".
To use the AN1 pin	Set the pins to "0001 <sub>B</sub> ".
To use the AN2 pin	Set the pins to "0010 <sub>B</sub> ".
To use the AN3 pin	Set the pins to "0011 <sub>B</sub> ".
To use the AN4 pin	Set the pins to "0100 <sub>B</sub> ".
To use the AN5 pin	Set the pins to "0101 <sub>B</sub> ".
To use the AN6 pin	Set the pins to "0110 <sub>B</sub> ".
To use the AN7 pin	Set the pins to "0111 <sub>B</sub> ".



● Checking the completion of conversion

The following two methods can be used to check whether conversion has been completed.

- Checking with the interrupt request flag bit (ADC1.ADI)

Interrupt request flag bit (ADI)	Meaning
The value read is "0".	A/D conversion completed with no interrupt request
The value read is "1".	A/D conversion completed with interrupt request generated

- Checking with the conversion flag bit (ADC1.ADMV)

Conversion flag bit (ADMV)	Setting
The value read is "0".	A/D conversion completed (suspended)
The value read is "1".	A/D conversion in progress

● Interrupt-related register

Use the following interrupt level setting register to set the interrupt level.

Interrupt source	Interrupt level setting register	Interrupt vector
8/10-bit A/D converter	Interrupt level register (ILR4) Address: 0007D <sub>H</sub>	#18 Address: 0FFD6 <sub>H</sub>

● Enabling, disabling, and clearing interrupts

To enable interrupts, use the interrupt request enable bit (ADC2.ADIE).

Control item	Interrupt request enable bit (ADIE)
To disable interrupt requests	Set the bit to "0".
To enable interrupt requests	Set the bit to "1".

To clear interrupt requests, use the interrupt request bit (ADC1.ADI).

Control item	Interrupt request bit (ADI)
To clear an interrupt request	Set the bit to "0". Or activate A/D.

# ***CHAPTER 25***

---

# ***LOW-VOLTAGE DETECTION RESET CIRCUIT***

**This chapter describes the functions and operations of the low-voltage detection reset circuit.**

- 25.1 Overview of Low-voltage Detection Reset Circuit
- 25.2 Configuration of Low-voltage Detection Reset Circuit
- 25.3 Pins of Low-voltage Detection Reset Circuit
- 25.4 Operations of Low-voltage Detection Reset Circuit

## 25.1 Overview of Low-voltage Detection Reset Circuit

---

**The low-voltage detection reset circuit monitors the power supply voltage and generates a reset signal if the voltage drops below the detection voltage level (available as an option to 5-V products only).**

---

### ■ Low-voltage Detection Reset Circuit

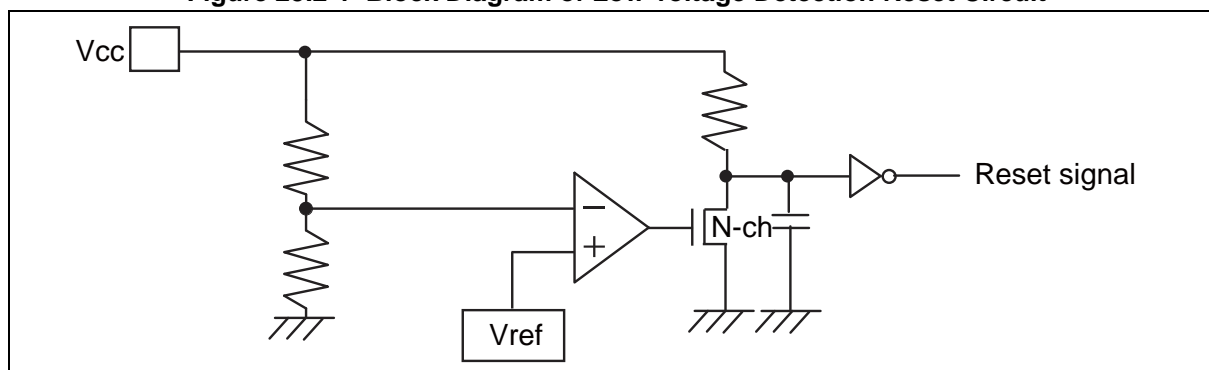
This circuit monitors the power supply voltage and generates a reset signal if the voltage drops below the detection voltage level. The circuit can be selected as an option to 5-V products only. Refer to the data sheet for details of the electrical characteristics.

## 25.2 Configuration of Low-voltage Detection Reset Circuit

Figure 25.2-1 is a block diagram of the low-voltage detection reset circuit.

### ■ Block Diagram of Low-voltage Detection Reset Circuit

Figure 25.2-1 Block Diagram of Low-voltage Detection Reset Circuit



## 25.3 Pins of Low-voltage Detection Reset Circuit

---

This section explains the pins of the low-voltage detection reset circuit.

---

### ■ Pins Related to Low-voltage Detection Reset Circuit

- Vcc pin

The low-voltage detection reset circuit monitors the voltage at this pin.

- Vss pin

This pin is a GND pin serving as the reference for voltage detection.

- $\overline{\text{RST}}$  pin

The low-voltage detection reset signal is output inside the microcontroller and to this pin.

However, for the model equipped with the clock supervisor function (see "1.2 Product Lineup of MB95110B/M Series" for details), the low-voltage detection reset signal is generated only in the microcontroller and not output to this pin.

## 25.4 Operations of Low-voltage Detection Reset Circuit

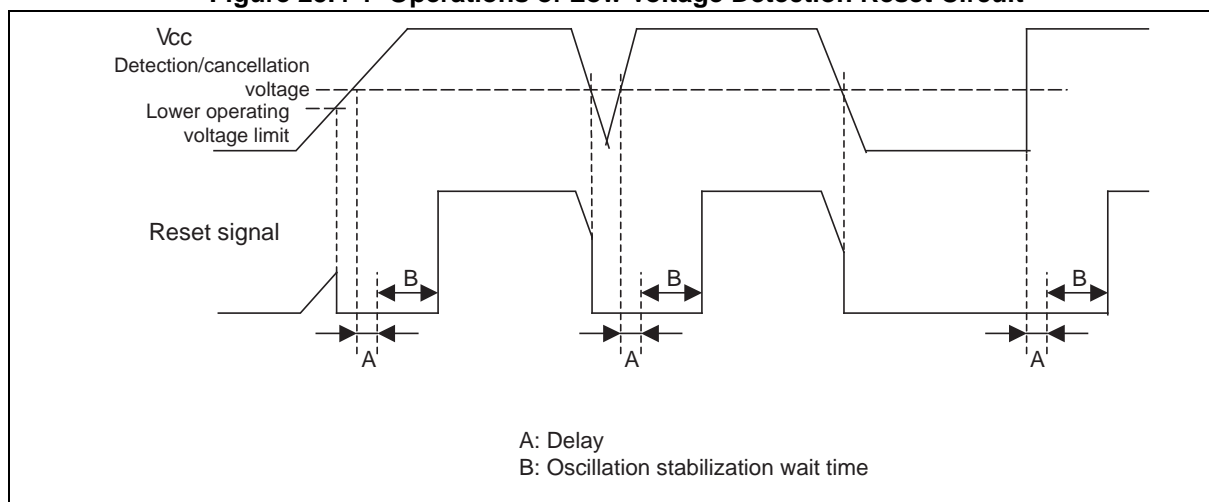
The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage.

### ■ Operations of Low-voltage Detection Reset Circuit

The low-voltage detection reset circuit generates a reset signal if the power supply voltage falls below the detection voltage. If the voltage is subsequently detected to have recovered, the circuit outputs a reset signal for the duration of the oscillation stabilization wait time to cancel the reset.

For details on the electrical characteristics, see the data sheet.

**Figure 25.4-1 Operations of Low-voltage Detection Reset Circuit**



### ■ Operations in Standby Mode

The low-voltage detection reset circuit remains operating even in standby modes (stop, sleep, sub clock, and watch modes).



# ***CHAPTER 26***

---

# ***CLOCK SUPERVISOR***

**This chapter describes the functions and operations of the clock supervisor.**

- 26.1 Overview of Clock Supervisor
- 26.2 Configuration of Clock Supervisor
- 26.3 Register of Clock Supervisor
- 26.4 Operations of Clock Supervisor
- 26.5 Notes on Using Clock Supervisor



## **26.1 Overview of Clock Supervisor**

---

**The clock supervisor prevents the situation which is out of control, when main clock and sub clock (only on dual clock products) oscillations have halted. This function switches to an CR clock generated in internal CR oscillator circuit, if main clock and sub clock oscillations have halted (this feature is optional to 5-V products).**

---

### **■ Overview of Clock Supervisor**

- The clock supervisor monitors the main clock and sub clock oscillations and generates an internal reset if it detects that the oscillation has halted. In this case, the clock supervisor switches to the internal CR clock (the clock frequency of the sub clock is equal to the CR clock frequency divided by 2).  
The reset source register (RSRR) can be used to determine whether a reset was triggered by the clock supervisor.
- A main clock oscillation halt is detected if the rising edge of the main clock is not detected for 4 CR clock cycles. The clock supervisor may detect incorrectly, if main clock is longer than 4 CR clock cycles.
- A sub clock oscillation halt is detected if the rising edge of the sub clock is not detected for 32 CR clock cycles. The clock supervisor may detect incorrectly, if sub clock is longer than 32 CR clock cycles.
- The clock supervisor can prohibit to monitor the main clock and sub clock respectively.
- If the sub clock is halted in the main clock mode, a reset does not occur immediately, but does occur after switching to the sub clock mode.  
Setting registers enable to prohibit the reset output.
- While the clock stops in main clock and sub clock stop modes, clock monitoring is disabled.
- This function can be selected as an option on 5-V products only.

---

Note:

Refer to the data sheet for the period and other details about the CR clock.

---

## 26.2 Configuration of Clock Supervisor

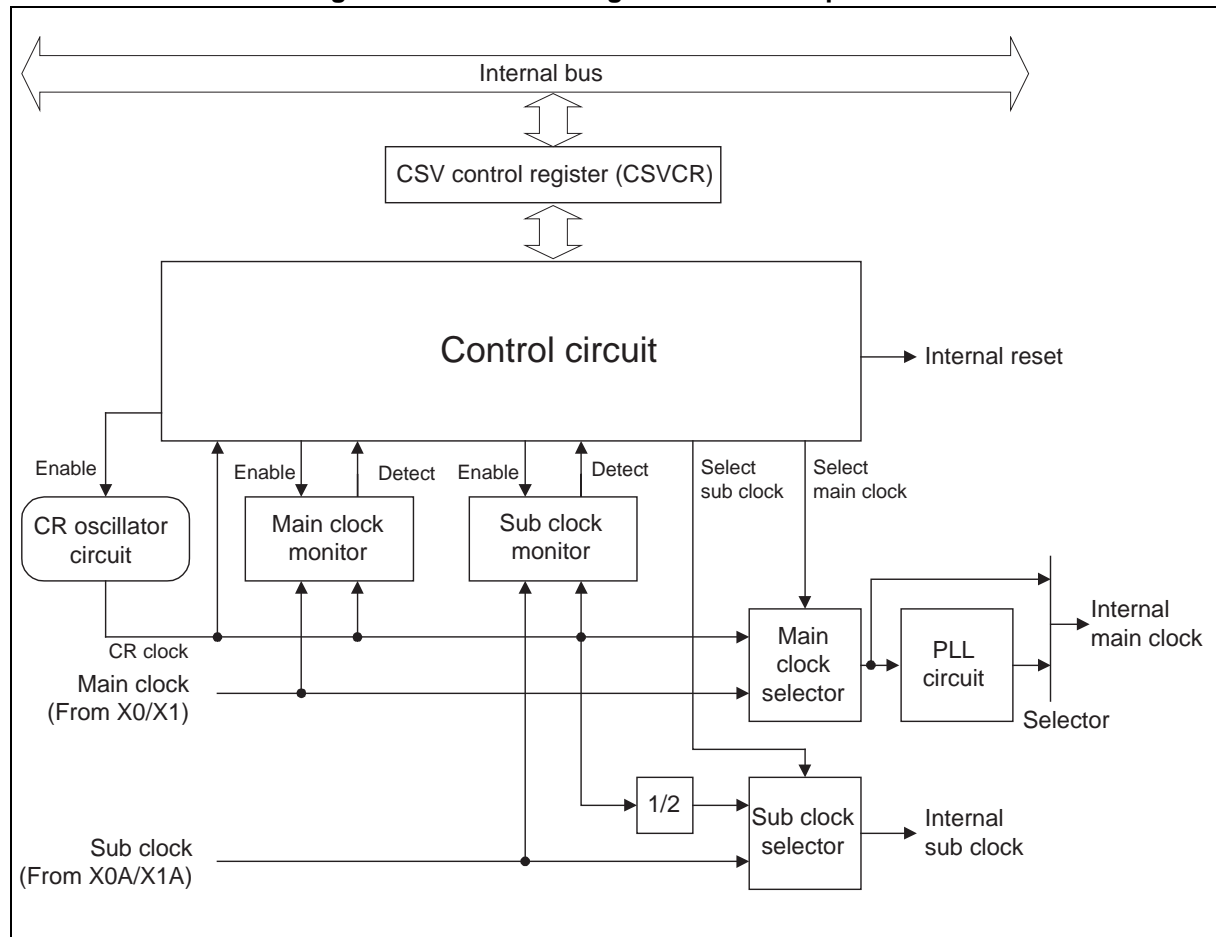
The clock supervisor consists of the following blocks:

- Control circuit
- CR oscillator circuit
- Main clock monitor
- Sub clock monitor
- Main clock selector
- Sub clock selector
- CSV control register (CSVCR)

### ■ Block Diagram of Clock Supervisor

Figure 26.2-1 shows a block diagram of the clock supervisor.

Figure 26.2-1 Block Diagram of Clock Supervisor



- **Control circuit**

This block controls the clocks, resets, and other settings based on the information in the CSV control register (CSVCR).

- **CR oscillator circuit**

This block is a internal CR oscillator circuit. The oscillation can be turned on or off via a control signal from the control circuit.

This also serves as an internal clock after a clock halt is detected.

- **Main clock monitor**

This block monitors whether the main clock halts.

- **Sub clock monitor**

This block monitors whether the sub clock halts.

- **Main clock selector**

This block outputs the CR clock as the internal main clock upon detection of a main clock halt.

- **Sub clock selector**

This block outputs the clock obtained by dividing the CR clock as the internal sub clock upon detection of a sub clock halt.

- **CSV control register (CSVCR)**

This block is used to control clock monitoring and CR clock and to check information on halt detection.

## 26.3 Register of Clock Supervisor

This section describes the clock supervisor registers.

### ■ Register of Clock Supervisor

Figure 26.3-1 shows the register of the clock supervisor.

**Figure 26.3-1 Clock Supervisor Register**

Clock supervisor control register (CSVCR)								
bit	7	6	5	4	3	2	1	0
Address 000FEA <sub>H</sub>	Reserv ed	MM	SM	RCE	MSVE	SSVE	SRST	Reserv ed
	R/W	R	R	R/W	R/W	R/W	R/W	R/W
R/W : Readable/writable								
R : Read only								

## 26.3.1 Clock Supervisor Control Register (CSVCR)

The clock supervisor control register (CSVCR) is used to enable the various functions and to check the status.

### ■ Clock Supervisor Control Register (CSVCR)

Figure 26.3-2 Clock Supervisor Control Register (CSVCR)

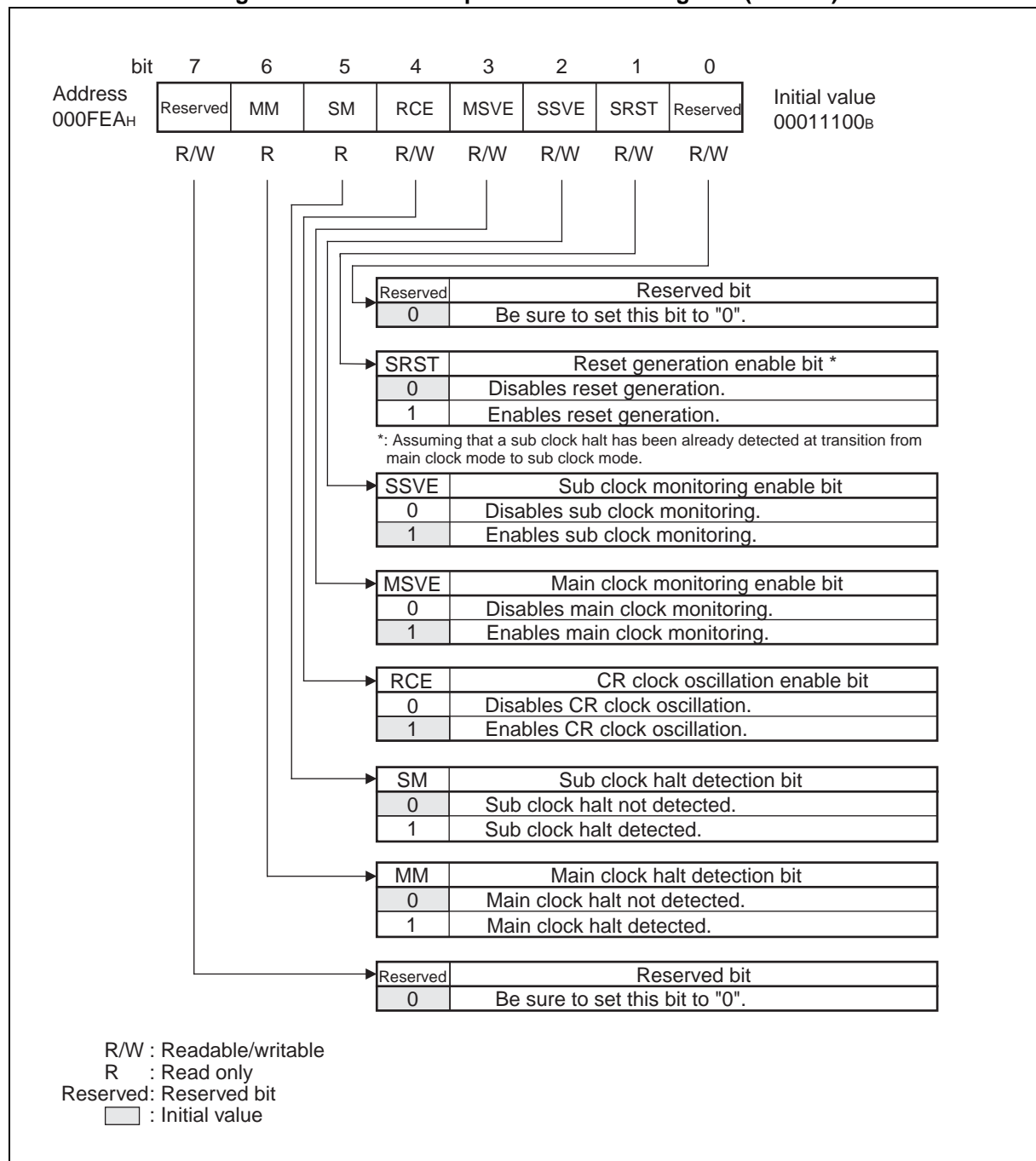


Table 26.3-1 Functions of Bits in Clock Supervisor Control Register (CSVCR)

Bit name		Function
bit7	Reserved bit	This bit is reserved. Write "0" to this bit. The read value is always "0".
bit6	MM: Main clock halt detection bit	This bit is read-only, and this bit indicates that a main clock oscillation halt has been detected. <b>When set to "0"</b> : The bit indicates that no main clock oscillation halt has been detected. <b>When set to "1"</b> : The bit indicates that main clock oscillation halt has been detected. Writing "1" to this bit does not affect the operation.
bit5	SM: Sub clock halt detection bit	This bit is read-only, and this bit indicates that a sub clock oscillation halt has been detected. <b>When set to "0"</b> : The bit indicates that no sub clock oscillation halt has been detected. <b>When set to "1"</b> : The bit indicates that sub clock oscillation halt has been detected. Writing "1" to this bit does not affect the operation.
bit4	RCE: CR clock oscillation enable bit	This bit enables CR oscillation. <b>When set to "0"</b> : The bit disables oscillation. <b>When set to "1"</b> : The bit enables oscillation (initial value). Before writing "0" to this bit, make sure that the clock monitor function has been disabled with the MM and SM bits set to "0".
bit3	MSVE: Main clock monitoring enable bit	This bit enables the monitoring of main clock oscillation. <b>When set to "0"</b> : The bit disables main clock monitoring. <b>When set to "1"</b> : The bit enables main clock monitoring. This bit is set to "1" only when a power-on reset occurs.
bit2	SSVE: Sub clock monitoring enable bit	This bit enables the monitoring of sub clock oscillation. <b>When set to "0"</b> : The bit disables sub clock monitoring. <b>When set to "1"</b> : The bit enables sub clock monitoring. This bit is set to "1" only when a power-on reset occurs.
bit1	SRST: Reset generation enable bit	This bit enables reset output upon transition to sub mode. <b>When set to "0"</b> : The bit prevents a reset upon transition to sub clock mode with the sub clock halted in main clock mode. <b>When set to "1"</b> : The bit causes a reset upon transition to sub clock mode with the sub clock halted in main clock mode.
bit0	Reserved bit	This bit is reserved. Write "0" to this bit. The read value is always "0".

**Note:**

When the power is turned on, the clock supervisor starts monitoring after the oscillation stabilization wait time for the main clock elapses. The oscillation stabilization wait time of the main clock must therefore be longer than the time required for the clock supervisor to start operating.

## 26.4 Operations of Clock Supervisor

---

**This section describes the operations of the clock supervisor.**

---

### ■ Operations of Clock Supervisor

The clock supervisor monitors the main clock and sub clock oscillations. If main clock and sub clock oscillations have halted, the device switches to an CR clock and generates a reset.

The following describes the operation in each clock mode.

#### ● Main clock oscillation halt in main clock mode

The clock supervisor detect that main clock oscillation has halted, if no rising edge is detected on the main clock for 4 CR clock cycles in main clock mode.

If a main clock halt is detected, a reset is generated and the main clock switches to the CR clock.

The clock supervisor may detect incorrectly, if main clock is a low speed (longer than 4 CR clock cycles). It results from using the CR clock for detecting that main clock oscillation have halted.

The clock supervisor does not detect the main clock during stop mode.

#### ● Sub clock oscillation halt in main clock mode (only on dual clock products)

In main clock mode, the condition used to detect the sub clock oscillation as having halted is that no rising edge is detected on the sub clock for 32 CR clock cycles.

Although no reset is generated immediately if a sub clock halt is detected in main clock mode, the sub clock switches to CR clock divided by two.

A reset can be generated when the device switches from main clock mode to sub clock mode with a sub clock oscillation halt detected, by setting the SRST bit in the clock supervisor control register (CSVCR).

As the CR clock is used to detect whether the sub clock has halted, a sub clock halt may be detected if the sub clock is set to a low speed (period longer than 32 CR clock cycles).

The clock supervisor does not detect the sub clock during the stop mode.

#### ● Sub clock oscillation halt in sub clock mode (only on dual clock products)

In sub clock mode, the condition used to detect the sub clock oscillation as having halted is that no rising edge is detected on the sub clock for 34 CR clock cycles.

If a sub clock halt is detected, a reset is generated and the device enters main clock mode. In this case, the sub clock switches to CR clock divided by two.

As the CR clock is used to detect whether the sub clock has halted, a sub clock halt may be detected if the sub clock is set to a low speed (period longer than 32 CR clock cycles).

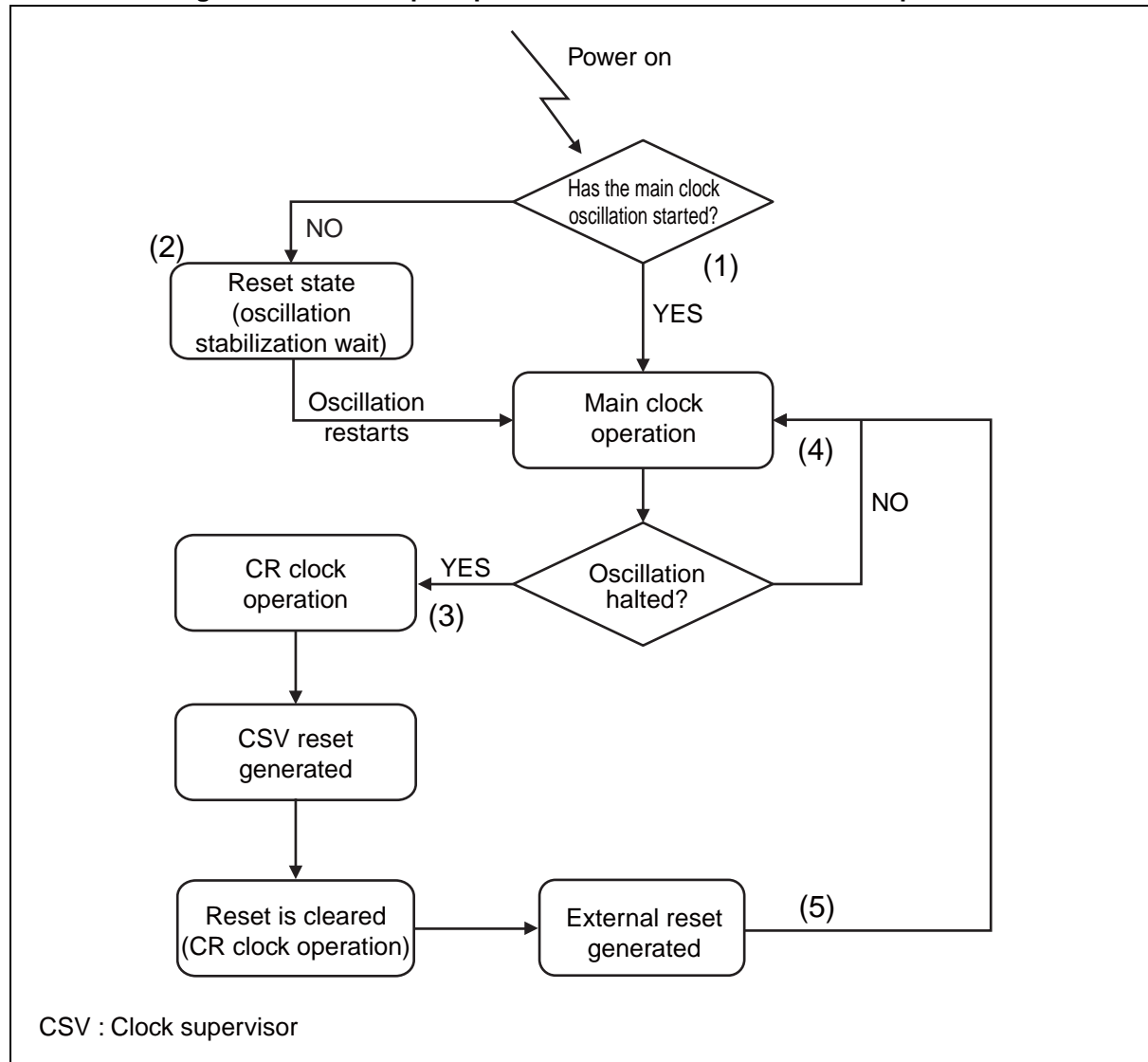
The clock supervisor does not detect the sub clock during the stop mode.

#### ● Main clock oscillation halt in sub clock mode (only on dual clock products)

In sub clock mode, the main clock oscillation remains halted and is therefore not detected by the clock supervisor.

## ■ Example Operation Flowchart for the Clock Supervisor

Figure 26.4-1 Example Operation Flowchart for the Clock Supervisor



- (1) After the power is turned on, the main clock operation starts after the oscillation stabilization wait time generated by the main clock oscillation has elapsed.
- (2) If the main clock halts at power on, the device remains in the reset state (oscillation stabilization wait state). The operation changes to the main clock, after the oscillation restarts and the oscillation stabilization wait time elapsed.
- (3) If an oscillation halt is detected during main clock operation, the operating clock is switched to the CR clock and a reset is generated.
- (4) If the main oscillation continues (oscillation does not halt), the device continues to run using the main clock.
- (5) If an external reset occurs during the CR clock operation, operation changes to the main clock. However, if the oscillation is halted at this time, another CSV reset is generated and the device returns to CR clock operation.

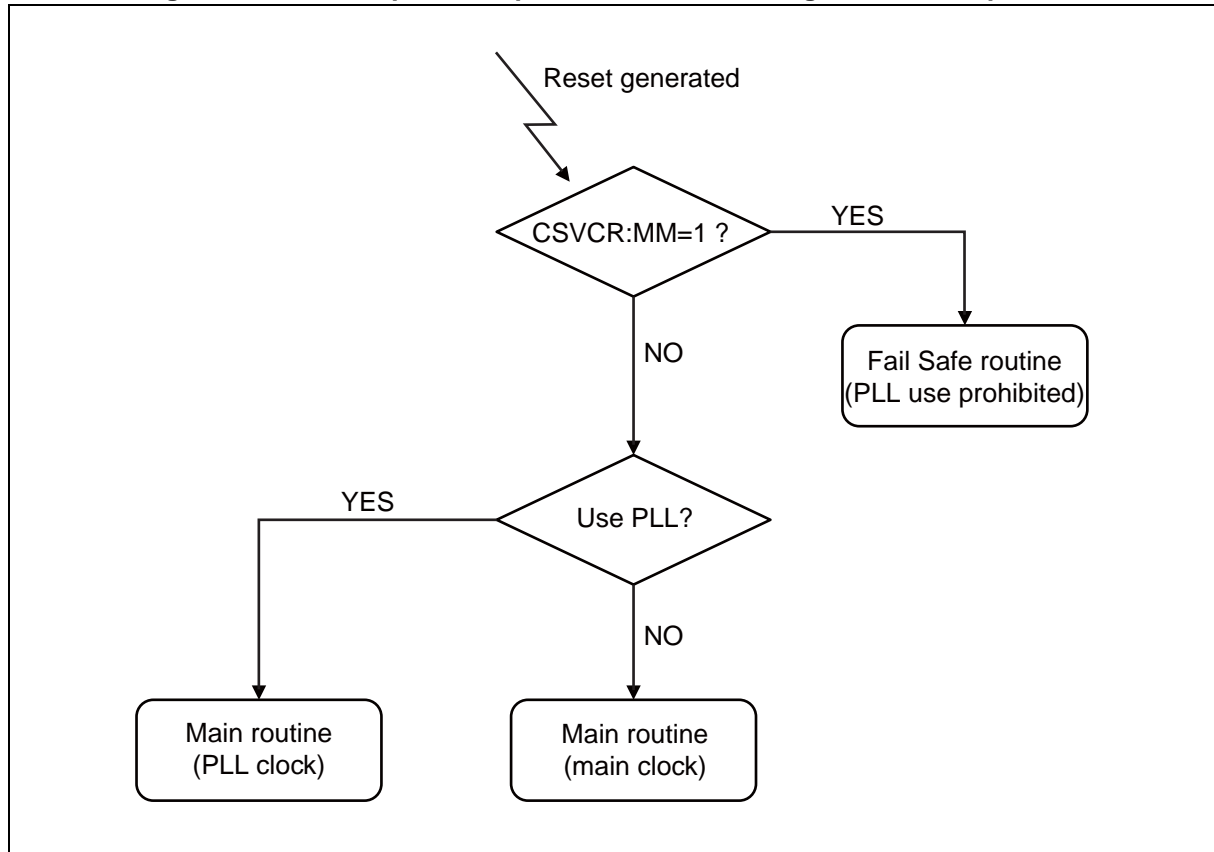


■ **Example Startup Flowchart when using the Clock Supervisor**

Inserting checking process of the main clock stop detection bit (CSVCR:MM) enables user programs to control the Fail Safe routine.

Figure 26.4-2 shows the example startup flowchart when using the clock supervisor.

**Figure 26.4-2 Example Startup Flowchart when using the Clock Supervisor**



## 26.5 Notes on Using Clock Supervisor

---

Take note of the following points when using the clock supervisor.

---

### ■ Notes on Using Clock Supervisor

Points to Note when using the Clock Supervisor

- Operation of the clock supervisor at power on  
When the power is turned on, the clock supervisor starts monitoring after the oscillation stabilization wait time for the main clock has elapsed. Therefore, unless the operation continues for longer than the oscillation stabilization wait time for the main clock, the clock supervisor will not operate.
- Transition to CR clock mode  
Do not turn on the PLL after changing to CR clock mode.  
As the frequency is below the lower limit for the input frequency of the PLL circuit, the PLL operation will not be guaranteed.
- Disabling the CR oscillation  
Do not use the CR oscillation enable bit (CSVCR:RCE) to disable the CR oscillation during CR clock mode.  
As this halts the internal clock, it may result in deadlock.
- Initializing the main clock halt detection bit  
The main clock halt detection bit (CSVCR:MM) is initialized by a power-on reset or external reset only. The bit is not initialized by the watchdog timer reset/software reset/CSV reset. Accordingly, the device remains in CR clock mode if one of these resets occurs during CR clock mode.



# **CHAPTER 27**

---

# **DUAL-OPERATION FLASH MEMORY**

**This chapter describes the functions and operations of 480-Kbit dual-operation flash memory.**

- 27.1 Overview of Dual-Operation Flash Memory
- 27.2 Sector/Bank Configuration of Flash Memory
- 27.3 Register of Flash Memory
- 27.4 Starting the Flash Memory Automatic Algorithm
- 27.5 Checking the Automatic Algorithm Execution Status
- 27.6 Flash Memory Program/Erase
- 27.7 Operation of Dual-Operation Flash Memory
- 27.8 Flash Security
- 27.9 Notes on Using Dual-Operation Flash Memory

## 27.1 Overview of Dual-Operation Flash Memory

Dual-operation flash memory is located from 1000<sub>H</sub> to FFFF<sub>H</sub> on the CPU memory map. The function of the flash memory interface circuit provides read access and program access from the CPU to flash memory.

Dual-operation flash memory consists of upper bank (16K bytes × 2 + 4K bytes × 4) and lower bank (4K bytes × 3) and erasing/programming and a read can be executed concurrently each bank.

### ■ Overview of Dual-Operation Flash Memory

The following methods can be used to program (write) and erase data into/from flash memory:

- Programming/erasing using a parallel writer
- Programming/erasing using a dedicated serial writer
- Programming/erasing by program execution

As flash memory by program execution can be programmed and erased by the instructions from the CPU via the flash memory interface circuit, you can efficiently reprogram (update) program code and data in flash memory with the device mounted on a circuit board.

Sector is composed of minimum 4K byte and small sector so that it is easy to be handled as the program/data area.

Data can be updated by executing a program not only in RAM but also in flash memory in dual-operation mode. An erase/program and a read can be executed concurrently in the different banks (the upper and lower banks).

The following combination of dual-operation flash memory are available.

Upper bank	Lower bank
Read	
Read	Programming/Sector Erasing
Programming/Sector Erasing	Read
Chip erase	

Bank programming/sector erasing cannot be executed during programming/erasing for another bank.

**■ Features of Dual-Operation Flash Memory**

- Sector configuration: 60K bytes × 8 bits (4K bytes × 4 + 16K bytes × 2 + 4K bytes × 3)
- Two-bank configuration, enabling simultaneous execution of an erase/program and a read
- Automatic program algorithm (Embedded Algorithm)
- Erase suspend/resume function integrated
- Detection of completion of programming/erasing using the data polling or toggle bit function
- Detection of completion of programming/erasing by CPU interrupts
- Compatible with JEDEC standard commands
- Capable of erasing data from specific sectors (any combination of sectors)
- Programming/erase count (minimum): 10,000 times
- Flash read cycle time (minimum): 1 machine cycle

**■ Flash memory program/erase**

- It is not possible to write to and read from the same bank of flash memory at the same time.
- To program/erase data into/from a bank in flash memory, execute either a relevant program in a different bank in the flash memory or the one copied from the flash memory to RAM, so that writing to the flash memory can be performed.
- By using dual-operation flash memory, programs can be executed on the flash memory and programming control using interrupt is enabled. In addition, programs do not need to be downloaded onto the RAM for execution when programming, and it is not necessary to take measures reducing the downloading time or turning OFF the power of the RAM data.

## 27.2 Sector/Bank Configuration of Flash Memory

This section explains the registers and the sector/bank configuration of flash memory.

### ■ Sector/Bank Configuration of Dual-Operation Flash Memory

Figure 27.2-1 shows the sector configuration of the dual-operation flash memory. The upper and lower addresses of each sector are given in the figure.

#### ● Sector Configuration

When the CPU accesses flash memory, SA1 to SA9 are located at 1000<sub>H</sub> to FFFF<sub>H</sub>.

#### ● Bank configuration

The flash memory consists of the lower bank from SA1 to SA3 and the upper bank from SA4 to SA9.

**Figure 27.2-1 Sector Configuration of Dual-Operation Flash Memory**

Flash memory	CPU address	Programmer address*	
SA1 (4K bytes)	1000 <sub>H</sub> 1FFF <sub>H</sub>	71000 <sub>H</sub> 71FFF <sub>H</sub>	Lower bank
SA2 (4K bytes)	2000 <sub>H</sub> 2FFF <sub>H</sub>	72000 <sub>H</sub> 72FFF <sub>H</sub>	
SA3 (4K bytes)	3000 <sub>H</sub> 3FFF <sub>H</sub>	73000 <sub>H</sub> 73FFF <sub>H</sub>	
SA4 (16K bytes)	4000 <sub>H</sub> 7FFF <sub>H</sub>	74000 <sub>H</sub> 77FFF <sub>H</sub>	Upper bank
SA5 (16K bytes)	8000 <sub>H</sub> BFFF <sub>H</sub>	78000 <sub>H</sub> 7BFFF <sub>H</sub>	
SA6 (4K bytes)	C000 <sub>H</sub> CFFF <sub>H</sub>	7C000 <sub>H</sub> 7CFFF <sub>H</sub>	
SA7 (4K bytes)	D000 <sub>H</sub> DFFF <sub>H</sub>	7D000 <sub>H</sub> 7DFFF <sub>H</sub>	
SA8 (4K bytes)	E000 <sub>H</sub> EFFF <sub>H</sub>	7E000 <sub>H</sub> 7EFFF <sub>H</sub>	
SA9 (4K bytes)	F000 <sub>H</sub> FFFF <sub>H</sub>	7F000 <sub>H</sub> 7FFFF <sub>H</sub>	

\*: The programmer address is equivalent to the CPU address, which is used when data is written to the flash memory using parallel programmer. When a parallel programmer is used for programming/erasing, this address is used for programming/erasing.

**MB95110B/M Series****27.3 Register of Flash Memory**

This section shows the register of the flash memory

### ■ Register of Flash Memory

**Figure 27.3-1 Register of Flash Memory**

Flash Memory Status Register (FSR)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072 <sub>H</sub>	—	—	RDYIRQ	RDY	Reserved	IRQEN	WRE	SSEN	000X0000 <sub>B</sub>
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W	
Flash memory sector write control register (SWRE0)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0073 <sub>H</sub>	—	—	—	—	—	—	SA9E	SA8E	00000000 <sub>B</sub>
	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	
Flash memory sector write control register (SWRE1)									
Address	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
0074 <sub>H</sub>	SA7E	SA6E	SA5E	SA4E	SA3E	SA2E	SA1E	Reserved	00000000 <sub>B</sub>
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W0	
R/W : Readable/writable (Read value is the same as write value) R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction) R/WX : Read only (Readable, writing has no effect on operation) R/W0 : Reserved bit (Write value is "0", read value is the same as write value) R0/WX : Undefined bit (Read value is "0", writing has no effect on operation) X : Indeterminate									



## 27.3.1 Flash Memory Status Register (FSR)

Figure 27.3-2 lists the functions of the flash memory status register (FSR).

### ■ Flash Memory Status Register (FSR)

Figure 27.3-2 Flash Memory Status Register (FSR)

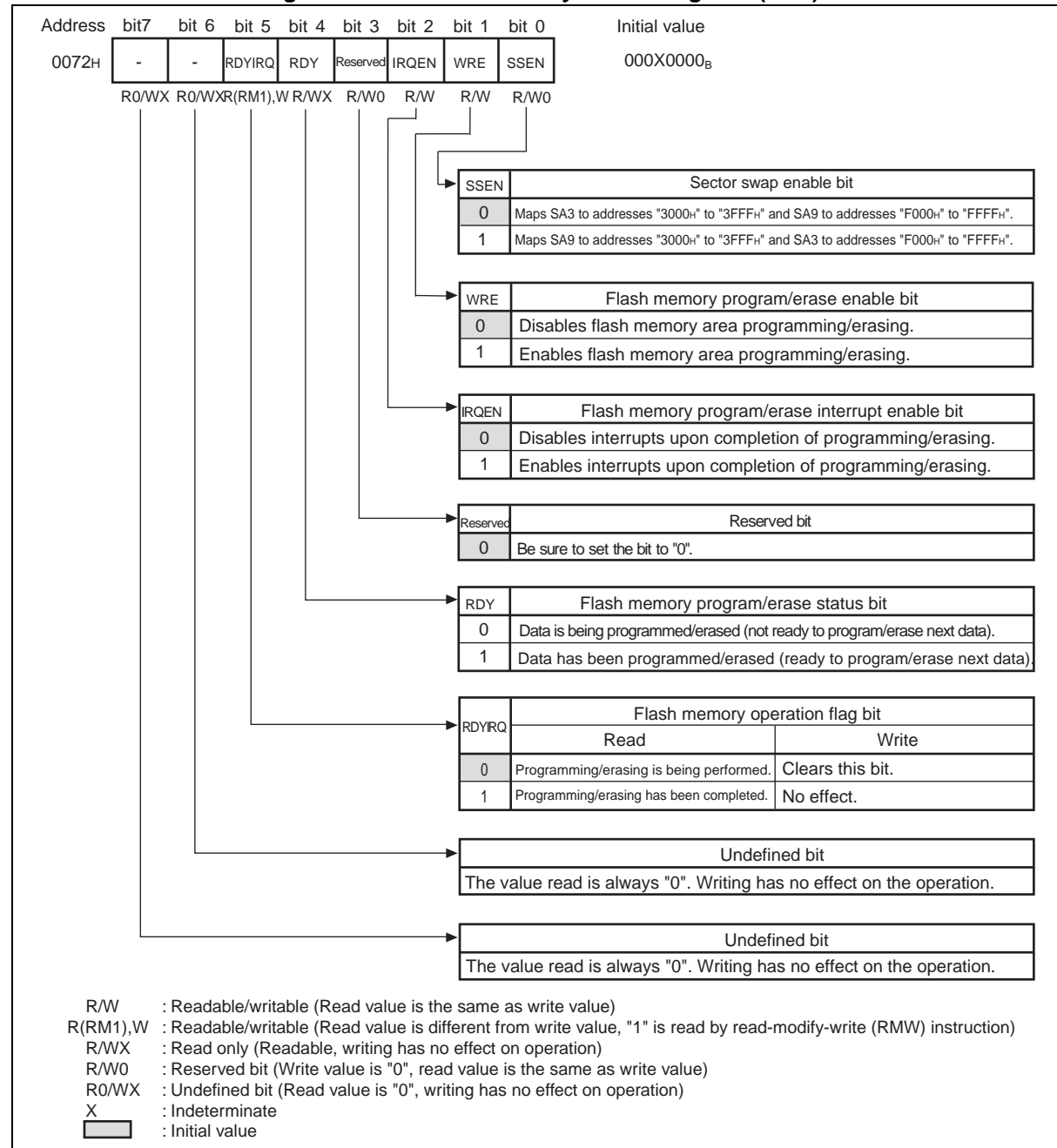
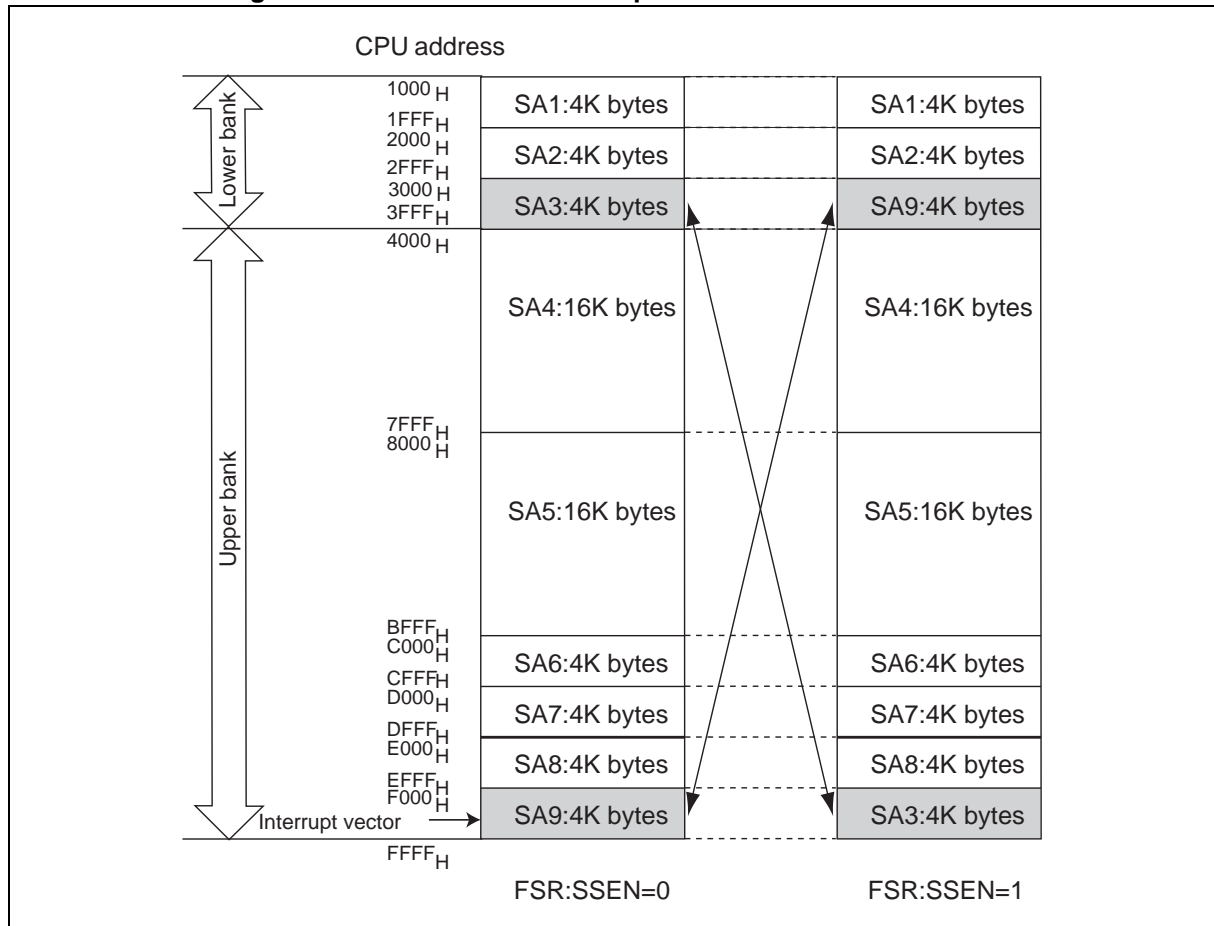


Table 27.3-1 Functions of Flash Memory Status Register (FSR)

Bit name		Function
bit7, bit6	-: Undefined bits	The value read is always "0". Writing has no effect on the operation.
bit5	RDYIRQ: Flash memory operation flag bit	<p>This bit shows the operating state of flash memory. The RDYIRQ bit is set to "1" upon completion of the flash memory automatic algorithm when flash memory programming/erasing is completed.</p> <ul style="list-style-type: none"> <li>An interrupt request occurs when the RDYIRQ bit is set to "1" if interrupts triggered by the completion of flash memory programming/erasing have been enabled (FSR: IRQEN=1).</li> <li>If the RDYIRQ bit is set to "0" when flash memory programming/erasing is completed, further flash memory programming/erasing is disabled.</li> </ul> <p><b>Setting the bit to "0"</b>: Clears the bit.  <b>Setting the bit to "1"</b>: Has no effect on the operation.  "1" is read from the bit whenever a read-modify-write (RMW) instruction is used.</p>
bit4	RDY: Flash memory program/erase status bit	<p>This bit shows the programming/erasing status of flash memory.</p> <ul style="list-style-type: none"> <li>Flash memory programming/erasing cannot be performed with the RDY bit set to "0".</li> <li>A read/reset command or sector-erase suspend command can be accepted even when the RDY bit contains "0". The RDY bit is set to "1" upon completion of programming/erasing.</li> <li>It takes a delay of two machine clock (MCLK) cycles after the issuance of a program/erase command for the RDY bit to be set to "0". Read this bit after, for example, inserting NOP twice after issuing the program/erase command.</li> </ul>
bit3	Reserved: Reserved bit	Be sure to set this bit to "0".
bit2	IRQEN: Flash memory program/erase interrupt enable bit	<p>This bit enables or disables the generation of interrupt requests in response to the completion of flash memory programming/erasing.</p> <p><b>Setting the bit to "0"</b>: Prevents an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1).</p> <p><b>Setting the bit to "1"</b>: Causes an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1).</p>
bit1	WRE: Flash memory program/erase enable bit	<p>This bit enables or disables the programming/erasing of data into/from the flash memory area. Set the WRE bit before invoking a flash memory program/erase command.</p> <p><b>Setting the bit to "0"</b>: Prevents a program/erase signal from being generated even when a program/erase command is input.</p> <p><b>Setting the bit to "1"</b>: Allows flash memory programming/erasing to be performed after a program/erase command is input.</p> <ul style="list-style-type: none"> <li>When flash memory is not to be programmed or erased, set the WRE bit to "0" to prevent it from being accidentally programmed or erased.</li> <li>To program into flash memory, set FSR:WRE to "1" to write-enable it and set the flash memory sector write control register (SWRE0/SWRE1). When FSR:WRE disables programming (contains "0"), write access to flash memory does not take place even though it is enabled by the flash memory write control register (SWRE0/SWRE1).</li> </ul>
bit0	SSEN: Sector Conversion Enable Bit	<p>The sector of an area where the interrupt vector exists is switched between the upper bank SA9 and lower bank SA3.</p> <p><b>Setting the bit to "0"</b>:SA3 is mapped to Address 3000<sub>H</sub> to 3FFF<sub>H</sub>, and SA9 is mapped to Address F000<sub>H</sub> to FFFF<sub>H</sub>. An interrupt vector exists in SA9.</p> <p><b>Setting the bit to "1"</b>:SA9 is mapped to Address 3000<sub>H</sub> to 3FFF<sub>H</sub>, and SA3 is mapped to Address F000<sub>H</sub> to FFFF<sub>H</sub>. An interrupt vector exists in SA3.</p> <p>Figure 27.3-3 shows the access sector map for the set value FSR:SSEN.</p>

**Figure 27.3-3 Access Sector Map for the Set Value FSR:SSEN**



## MB95110B/M Series

### 27.3.2 Flash Memory Sector Write Control Registers (SWRE0/SWRE1)

The flash memory sector write control registers (SWRE0/SWRE1) exist in the flash memory interface to be used to set the flash memory accidental write protect feature.

#### ■ Flash Memory Sector Write Control Registers (SWRE0/SWRE1)

The flash memory sector write control registers (SWRE0/SWRE1) contain the bits to enable/disable programming data into individual sectors (SA1 to SA9). The initial value of each bit is "0" to disable programming. Writing 1 to one of the bits enables programming data into the corresponding sector. Writing 0 to it prevents an accidental write from being executed to the sector. Once you have written 0 to the bit, therefore, you cannot write to the sector even though you write 1 to the bit. You have to reset the bit before you can write to the sector again.

**Figure 27.3-4 Flash Memory Sector Write Control Registers (SWRE0/SWRE1)**

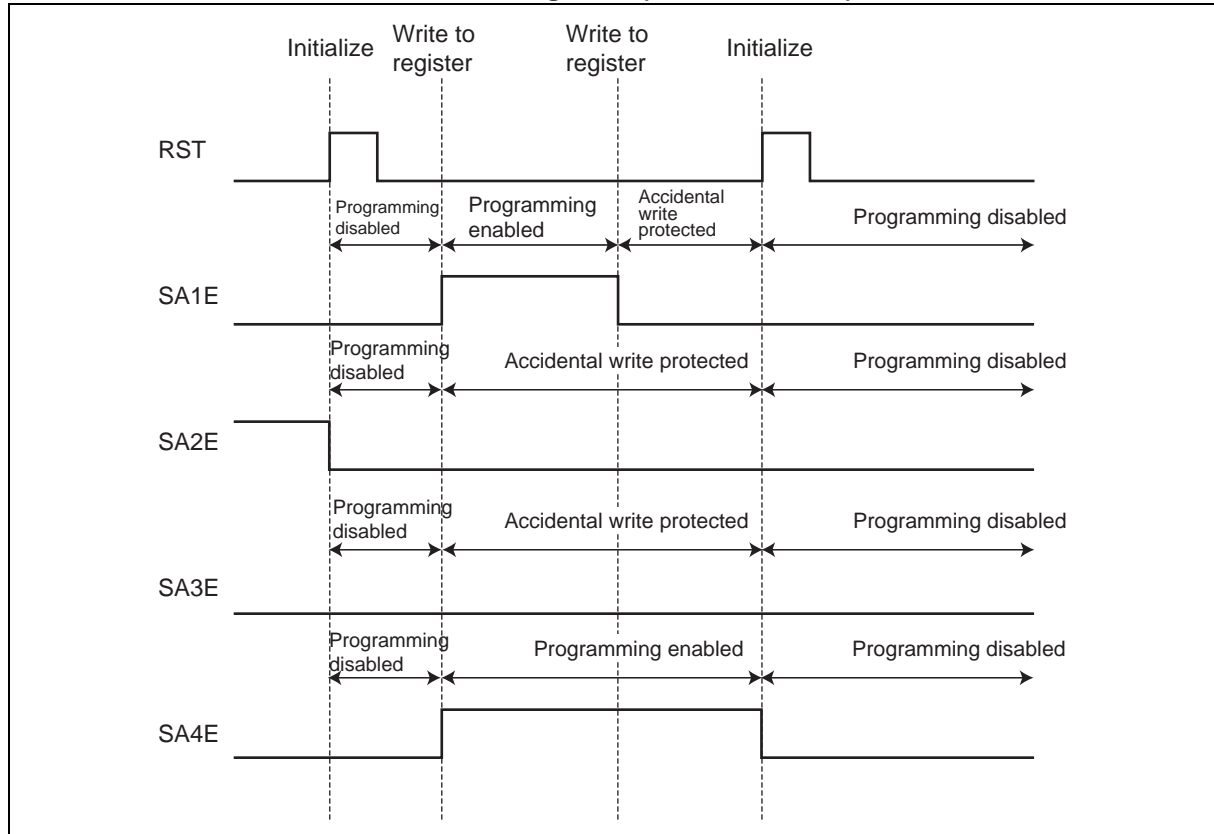
SWRE0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
Address	—	—	—	—	—	—	SA9E	SA8E	00000000 <sub>B</sub>
0073 <sub>H</sub>	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R0/WX	R/W	R/W	
SWRE1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	Initial value
Address	SA7E	SA6E	SA5E	SA4E	SA3E	SA2E	SA1E	Reserved	00000000 <sub>B</sub>
0074 <sub>H</sub>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W1	
<p>R/W : Readable/writable (Read value is the same as write value)  R/W1 : Reserved bit (Write value is "1", read value is the same as write value)  R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)</p>									

To these registers, be sure to write data in words. No bit manipulation instruction can be used for setting their bits.

**Table 27.3-2 Functions of Flash Memory Sector Write Control Registers (SWRE0/SWRE1)**

Register name	Bit name		Function																	
SWRE0	bit7 to bit2	-: Undefined bits	The value read is always "0". Writing has no effect on the operation.																	
	bit1, bit0	SA9E to SA1E: Programming function setup bits	These bits are used to set the accidental write function for the individual sectors of the flash memory. Writing "1" to one of these bits enables programming into the corresponding sector. Writing "0" to the bit write defect protects that sector (prevents an accidental write to the sector). Resetting the bit initializes it to 0 (programming disabled).																	
SWRE1	bit15 to bit9		Programming function setup bits and corresponding flash sectors																	
			<table><tr><th>Bit name</th><th>Corresponding sector in flash memory</th></tr><tr><td>SA9E</td><td>SA9</td></tr><tr><td>SA8E</td><td>SA8</td></tr><tr><td>SA7E</td><td>SA7</td></tr><tr><td>SA6E</td><td>SA6</td></tr><tr><td>SA5E</td><td>SA5</td></tr><tr><td>SA4E</td><td>SA4</td></tr><tr><td>SA3E</td><td>SA3</td></tr><tr><td>SA2E</td><td>SA2</td></tr><tr><td>SA1E</td><td>SA1</td></tr></table> <p><b>Programming disabled:</b> State of "0". The register corresponding to each sector can be set (to "1") to enable programming into that sector, with no "0" written in the flash memory sector write control registers (SWRE0/SWRE1). (State existing immediately after a reset.)</p> <p><b>Programming enabled:</b> State of "1". Data can be programmed into the corresponding sector.</p> <p><b>Accidental write protected:</b> State of "0".The register corresponding to each sector cannot be set (to "1") to enable programming into that sector even by writing "1" to it with "0" written in the flash memory sector write control registers (SWRE0/SWRE1).</p>	Bit name	Corresponding sector in flash memory	SA9E	SA9	SA8E	SA8	SA7E	SA7	SA6E	SA6	SA5E	SA5	SA4E	SA4	SA3E	SA3	SA2E
Bit name	Corresponding sector in flash memory																			
SA9E	SA9																			
SA8E	SA8																			
SA7E	SA7																			
SA6E	SA6																			
SA5E	SA5																			
SA4E	SA4																			
SA3E	SA3																			
SA2E	SA2																			
SA1E	SA1																			
	bit8	Reserved: Reserved bit	Be sure to set this bit to "1".																	

**Figure 27.3-5 Examples of Flash Memory Programming-Disabled, Programming-Enabled, and Accidental Write Protected States Depending on Flash Memory Sector Write Control Registers (SWRE0/SWRE1)**



**Programming disabled:**

State of "0". The register corresponding to each sector can be set (to "1") to enable programming into that sector, with no "0" written in the flash memory sector write control registers (SWRE0/SWRE1). (State existing immediately after a reset.)

**Programming enabled:**

State of "1". Data can be programmed into the corresponding sector.

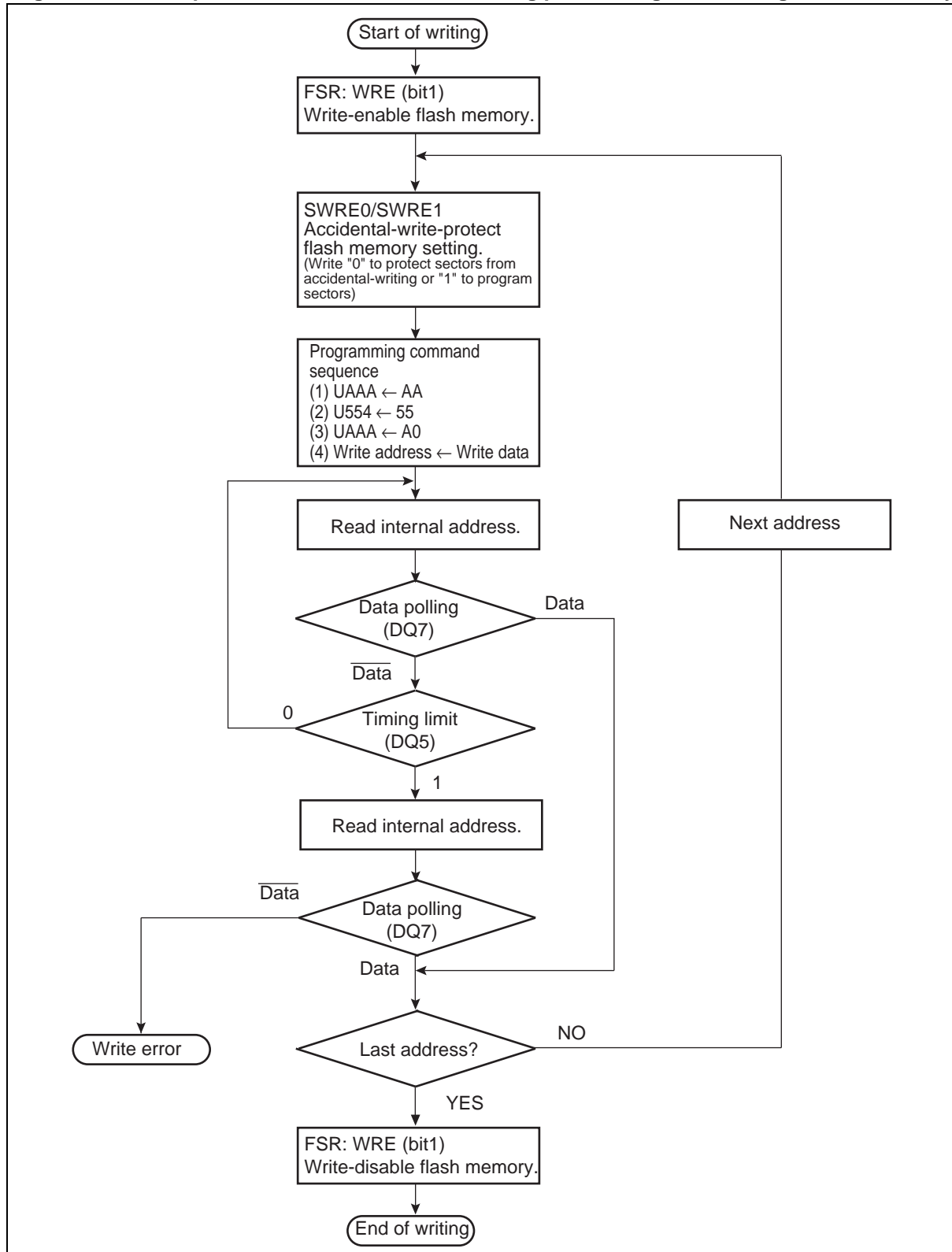
**Accidental write protected:**

State of "0". The register corresponding to each sector cannot be set (to "1") to enable programming into that sector even by writing "1" to it with "0" written in the flash memory sector write control registers (SWRE0/SWRE1).

## ■ Flash Memory Sector Write Control Registers (SWRE0/SWRE1) Setup Flowchart

Set the FSR:WRE bit and program-enable or accidental write protect each sector by setting the corresponding bit in the flash memory sector write control register (SWRE0/SWRE1) to "1" or "0", respectively.

**Figure 27.3-6 Sample Procedure for Accidental-writing protect/Program-enabling of Flash Memory**



**■ Note on Setting the FSR:WRE**

To program into flash memory, set FSR:WRE to "1" to write-enable it and set the flash memory sector write control registers (SWRE0/SWRE1). When FSR:WRE disables programming (contains "0"), write access to flash memory does not take place even though it is enabled by the flash memory sector write control registers (SWRE0/SWRE1).



## 27.4 Starting the Flash Memory Automatic Algorithm

There are four types of commands that invoke the flash memory automatic algorithm: read/reset, write (program), chip-erase, and sector-erase. The sector erase command is capable of suspending and resuming.

### ■ Command Sequence Table

Table 27.4-1 lists the command sequence table.

**Table 27.4-1 Command Sequence Table**

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	Address	Data
Read/reset*	1	F <sub>X</sub> XX <sub>H</sub>	F0 <sub>H</sub>	-	-	-	-	-	-	-	-	-	-
Read/reset*	4	UAAA <sub>H</sub>	AA <sub>H</sub>	U554 <sub>H</sub>	55 <sub>H</sub>	UAAA <sub>H</sub>	F0 <sub>H</sub>	RA	RD	-	-	-	-
Programming	4	UAAA <sub>H</sub>	AA <sub>H</sub>	U554 <sub>H</sub>	55 <sub>H</sub>	UAAA <sub>H</sub>	A0 <sub>H</sub>	PA	PD	-	-	-	-
Chip erasing	6	XAAA <sub>H</sub>	AA <sub>H</sub>	X554 <sub>H</sub>	55 <sub>H</sub>	XAAA <sub>H</sub>	80 <sub>H</sub>	XAAA <sub>H</sub>	AA <sub>H</sub>	X554 <sub>H</sub>	55 <sub>H</sub>	XAAA <sub>H</sub>	10 <sub>H</sub>
Sector erasing	6	UAAA <sub>H</sub>	AA <sub>H</sub>	X554 <sub>H</sub>	55 <sub>H</sub>	UAAA <sub>H</sub>	80 <sub>H</sub>	UAAA <sub>H</sub>	AA <sub>H</sub>	U554 <sub>H</sub>	55 <sub>H</sub>	SA	30 <sub>H</sub>
Sector erasing suspended		Entering address "UXXX <sub>H</sub> " and data "B0 <sub>H</sub> " suspends erasing during sector erasing.											
Sector erase resume		Entering address "UXXX <sub>H</sub> " and data "30 <sub>H</sub> " resumes erasing after erasing is suspended during sector erasing.											

- RA : Read address
  - PA : Write (program) address
  - SA : Sector address (specify arbitrary one address in sector)
  - RD : Read data
  - PD : Write (program) data
  - U : Upper 4 bits same as RA, PA, and SA
  - F<sub>X</sub> : FF/FE
  - X : Arbitrary address
- \*: Both of the two types of read/reset command can reset the flash memory to read mode.

#### Notes:

- Addresses in the table are the values in the CPU memory map. All addresses and data are hexadecimal values. However, "X" is an arbitrary value.
- Address "U" in the table is not arbitrary, whose four bits (bit15 to bit12) must have the same value as RA, PA, and SA.  
Example: If RA = C48E<sub>H</sub>, U = C; If PA = 1024<sub>H</sub>, U=1  
If SA = 3000<sub>H</sub>, U = 3
- The chip erase command is accepted only when all sectors have been program-enabled. The chip erase command is ignored if the bit for any sector in the flash memory sector write control registers (SWRE0/SWRE1) has been set to "0" (to program-disable or accidental-write protect the sector).

**■ Notes on Issuing Commands**

Pay attention to the following points when issuing commands in the command sequence table:

Program-enable each required sector before issuing the first command.

The upper address U bits (bit15 to bit12) used when commands are issued must have the same value as RA, PA, and SA, from the first command on.

If the above measures are not followed, commands are not recognized normally. Execute a reset to initialize the command sequencer in the flash memory.

## 27.5 Checking the Automatic Algorithm Execution Status

As the flash memory uses the automatic algorithm for a process flow for programming/erasing, you can check its internal operating status with hardware sequence flags.

### ■ Hardware Sequence Flag

#### ● Overview of hardware sequence flag

The hardware sequence flag consists of the following 4-bit outputs:

- Data Polling Flag (DQ7)
- Toggle Bit Flag (DQ6)
- Execution Time-out Flag (DQ5)
- Sector Erase Timer Flag (DQ3)

The hardware sequence flags tell whether the write (program), chip-erase, or sector-erase command has been terminated and whether an erase code write can be performed.

You can reference hardware sequence flags by read access to the address of each relevant sector in flash memory after setting a command sequence. Note, however, that hardware sequence flags are output only for the bank on a command-issued side.

Table 27.5-1 shows the bit allocation of the hardware sequence flags.

**Table 27.5-1 Bit Allocation of Hardware Sequence Flags**

Bit No.	7	6	5	4	3	2	1	0
Hardware sequence flag	DQ7	DQ6	DQ5	—	DQ3	—	—	—

- To know whether the automatic write, chip-erase, or sector-erase command is being executed or has been terminated, check the hardware sequence flags or the flash memory program/erase status bit in the flash memory status register (FSR:RDY). After programming/erasing is terminated, flash memory returns to the read/reset state.
- When creating a write/erase program, read data after checking the termination of automatic writing/erasing with the DQ7, DQ6, DQ5, and DQ3 flags.
- The hardware sequence flags can also be used to check whether the second sector erase code write and later are in effect.

● Explanation of hardware sequence flag

Table 27.5-2 lists the functions of the hardware sequence flag.

**Table 27.5-2 List of Hardware Sequence Flag Functions**

State		DQ7	DQ6	DQ5	DQ3
State transition during normal operation	Programming → Programming completed (when write address has been specified)	$\overline{DQ7} \rightarrow$ DATA: 7	Toggle → DATA: 6	0 → DATA: 5	0 → DATA: 3
	Chip/sector erasing → Erasing completed	0 → 1	Toggle → Stop	0 → 1	1
	Sector erasing wait → Erasing started	0	Toggle	0	0 → 1
	Erasing → Sector erasing suspended (Sector being erased)	0 → 1	Toggle → 1	0	1 → 0
	Sector erasing suspended → Erasing resumed (Sector being erased)	1 → 0	1 → Toggle	0	0 → 1
	Sector erasing being suspended (Sector not being erased)	DATA: 7	DATA: 6	DATA: 5	DATA: 3
Abnormal operation	Write	$\overline{DQ7}$	Toggle	1	0
	Chip/sector erasing	0	Toggle	1	1

## 27.5.1 Data Polling Flag (DQ7)

The data polling flag (DQ7) is a hardware sequence flag used to indicate that the automatic algorithm is being executing or has been completed using the data polling function.

### ■ Data Polling Flag (DQ7)

Table 27.5-3 and Table 27.5-4 show the state transition of the data polling flag (during normal operation) and the state transition of the data polling flag (during abnormal operation) respectively.

**Table 27.5-3 State Transition of Data Polling Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip/sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ7	$\overline{DQ7} \rightarrow \text{DATA: } 7$	0→1	0	0→1	1→0	DATA: 7

**Table 27.5-4 State Transition of Data Polling Flag (During Abnormal Operation)**

Operating state	Write	Chip/sector erasing
DQ7	$\overline{DQ7}$	0

#### ● At programming

When read access takes place during execution of the automatic write algorithm, the flash memory outputs the inverted value of bit7 in the last data written to DQ7.

If read access takes place on completion of the automatic write algorithm, the flash memory outputs bit7 of the value read from the read-accessed address to DQ7.

#### ● At chip/sector erasing

When read access is made to the sector currently being erased during execution of the chip/sector erase automatic algorithm, bit7 of flash memory outputs "0". Bit7 of flash memory outputs "1" upon completion of chip/sector erasing.

**● At sector erasing suspension**

- When read access takes place with a sector-erase operation suspended, the flash memory outputs "1" to DQ7 if the read address is the sector being erased. If not, the flash memory outputs bit7 (DATA:7) of the value read from the read address to DQ7.
- Referring the data polling flag (DQ7) together with the toggle bit flag (DQ6) permits a decision on whether flash memory is in the sector erase suspended state or which sector is being erased.

---

**Note:**

Once the automatic algorithm has been started, read access to the specified address is ignored. Data reading is allowed after the data polling flag (DQ7) is set to "1". Data reading after the end of the automatic algorithm should be performed following read access made to confirm the completion of data polling.

---

## 27.5.2 Toggle Bit Flag (DQ6)

The toggle bit flag (DQ6) is a hardware sequence flag used to indicate that the automatic algorithm is being executed or has been completed using the toggle bit function.

### ■ Toggle Bit Flag (DQ6)

Table 27.5-5 and Table 27.5-6 show the state transition of the toggle bit flag (during normal operation) and the state transition of the toggle bit flag(during abnormal operation) respectively.

**Table 27.5-5 State Transition of Toggle Bit Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip/ sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ6	Toggle → DATA: 6	Toggle→Stop	Toggle	Toggle→1	1→Toggle	DATA: 6

**Table 27.5-6 State Transition of Toggle Bit Flag (During Abnormal Operation)**

Operating state	Write	Chip/sector erasing
DQ6	Toggle	Toggle

#### ● At programming and chip/sector erasing

When read access is made continuously during execution of the automatic write algorithm or chip-erase/sector-erase automatic algorithm, the flash memory toggles the output between "1" and "0" at each read access.

When read access is made continuously after the automatic write algorithm or chip-erase/sector-erase automatic algorithm is terminated, the flash memory outputs bit6 (DATA:6) of the value read from the read address at each read access.

#### ● At sector erasing suspension

When read access is made with a sector erase operation suspended, the flash memory outputs "1" if the read address is the sector being erased. If not, the flash memory outputs bit6 (DATA:6) of the value read from the read address.

---

**Note:**

When using dual-operation flash memory (flash memory write control program is executed from flash memory), the toggle bit flag (DQ6) cannot be used to check the status during writing or erasing. Please refer to the notes in "27.9 Notes on Using Dual-Operation Flash Memory" when writing your program.

This precaution does not apply if running the flash memory write control program from RAM.

---



## 27.5.3 Execution Time-out Flag (DQ5)

The Execution Time-out Flag (DQ5) is a hardware sequence flag indicating that the automatic algorithm has been executed beyond the specified time (required for programming/erasing) internal to the flash memory.

### ■ Execution Time-out Flag (DQ5)

Table 27.5-7 and Table 27.5-8 show the state transition of the execution time-out flag (during normal operation) and the state transition of the execution time-out flag (during abnormal operation) respectively.

**Table 27.5-7 State Transition of Execution Time-out Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip/ sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ5	0 → DATA: 5	0 → 1	0	0	0	DATA: 5

**Table 27.5-8 State Transition of Execution Time-out Flag (During Abnormal Operation)**

Operating state	Write	Chip/sector erasing
DQ5	1	1

#### ● At programming and chip/sector erasing

When read access is made with the write or chip-erase/sector-erase automatic algorithm invoked, the flag outputs "0" when the algorithm execution time is within the specified time (required for programming/erasing) or "1" when it exceeds that time.

The execution time-out flag (DQ5) can be used to check whether programming/erasing has succeeded or failed regardless of whether the automatic algorithm has been running or terminated. When the execution time-out flag (DQ5) outputs "1", it indicates that programming has failed if the automatic algorithm is still running for the data polling or toggle bit function.

If an attempt is made to write "1" to a flash memory address holding "0", for example, the flash memory is locked, preventing the automatic algorithm from being terminated and valid data from being output from the data polling flag (DQ7). As the toggle bit flag (DQ6) does not stop toggling, the time limit is exceeded and the execution time-out flag (DQ5) outputs "1". The state in which the execution time-out flag (DQ5) outputs "1" means that the flash memory has not been used correctly; it does not mean that the flash memory is defective. When this state occurs, execute the reset command.

## 27.5.4 Sector Erase Timer Flag (DQ3)

The sector erase timer flag (DQ3) is a hardware sequence flag used to indicate whether the flash memory is waiting for sector erasing after the sector erase command has started.

### ■ Sector Erase Timer Flag (DQ3)

Table 27.5-9 and Table 27.5-10 show the state transition of the sector erase timer flag (during normal operation) and the state transition of the sector erase timer flag (during abnormal operation) respectively.

**Table 27.5-9 State Transition of Sector Erase Timer Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip/ sector erasing → Erasing completed	Sector erasing wait → Erasing started	Sector erasing → Sector erasing suspended (Sector being erased)	Sector erasing suspended → Erasing resumed (Sector being erased)	Sector erasing being suspended (Sector not being erased)
DQ3	0 → DATA: 3	1	0 → 1	1 → 0	0 → 1	DATA: 3

**Table 27.5-10 State Transition of Sector Erase Timer Flag (During Abnormal Operation)**

Operating state	Write	Chip/sector erasing
DQ3	0	1

#### ● At sector erasing

- When read access is made after the sector erase command has been started, the sector erase timer flag (DQ3) outputs "0" within the sector erasing wait period. The flag outputs "1" if the sector erase wait period has been exceeded.
- When the sector erase timer flag (DQ3) is "1", sector erasing is being performed if the erase algorithm shows running for the data polling or toggle bit function (with DQ7 holding 0 and DQ6 toggling the output). If any command other than the sector erase suspend command is set subsequently, it is ignored until sector erasing is terminated.
- If the sector erase timer flag (DQ3) is "0", flash memory can accept the sector erase command. Before programming the sector erase command into flash memory, make sure that the sector erase timer flag (DQ3) is "0". If the flag is "1", flash memory may not accept the sector erase suspended command.

#### ● At sector erasing suspension

When read access is made with a sector erase operation suspended, the flash memory outputs "1" if the read address is the sector being erased. If not, the flash memory outputs bit3 (DATA:3) of the value read from the read address.

## **27.6 Flash Memory Program/Erase**

---

**This section describes the individual procedures for flash memory reading/resetting, programming, chip-erasing, sector-erasing, sector-erase suspending, and sector-erase resuming by entering their respective commands to invoke the automatic algorithm.**

---

### **■ Details of Programming/Erasing Flash Memory**

The automatic algorithm can be invoked by writing the read/reset, program, chip-erase, sector-erase, sector-erase suspend, and sector-erase resume command sequence to flash memory from the CPU. Writing command sequence to flash memory from the CPU must always be performed continuously. The termination of the automatic algorithm can be checked by the data polling function. After the automatic algorithm terminates normally, the flash memory returns to the read/reset state.

The individual operations are explained in the following order:

- Enter read/reset state.
- Program data.
- Erase all data (chip-erase).
- Erase arbitrary data (sector erase).
- Suspend sector erasing.
- Resume sector erasing.

## 27.6.1 Placing Flash Memory in the Read/Reset State

---

**This section explains the procedure for entering the read/reset command to place flash memory in the read/reset state.**

---

### ■ Placing Flash Memory in the Read/Reset State

- To place flash memory in the read/reset state, send the read/reset command in the command sequence table continuously from the CPU to flash memory.
- The read/reset command is available in two different command sequences: one involves a single bus operation and the other involves four bus operations, which are essentially the same.
- Since the read/reset state is the initial state of flash memory, the flash memory always enters this state after the power is turned on and at the normal termination of a command. The read/reset state is also described as the wait state for command input.
- In the read/reset state, read access to flash memory enables data to be read. As is the case with masked ROM, program access from the CPU can be made.
- Read access to flash memory does not require the read/reset command. If a command is not terminated normally, use the read/reset command to initialize the automatic algorithm.

## **27.6.2 Programming Data into Flash Memory**

---

**This section explains the procedure for entering the write (program) command to program data into flash memory.**

---

### **■ Programming Data into Flash Memory**

- To start the automatic algorithm for programming data into flash memory, send the program command in the command sequence table continuously from the CPU to flash memory.
- Upon completion of data programming to a target address in the fourth cycle, the automatic algorithm is activated to start automatic programming.

#### **● How to specify addresses**

Programming (writing) can be performed even in any order of addresses or across a sector boundary. Data written by a single program command is only one byte.

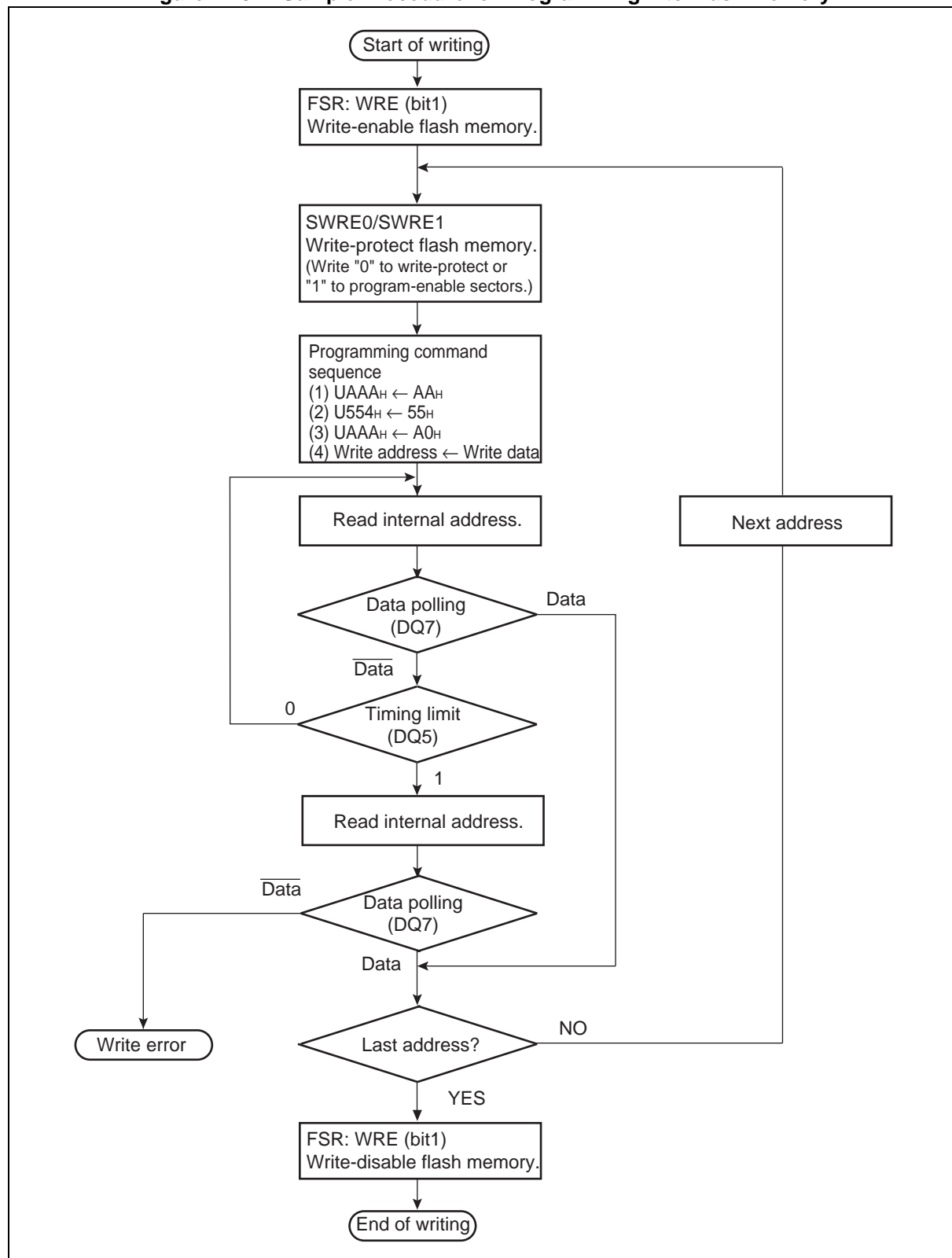
#### **● Notes on programming data**

- Bit data cannot be returned from "0" to "1" by programming. When bit data "1" is programmed to bit data "0", the data polling function (DQ7) or toggle operation (DQ6) is not terminated, the flash memory element is determined to be defective, and the execution time-out flag (DQ5) detects an error to indicate that the specified programming time has been exceeded.  
When data is read in the read/reset state, the bit data remains "0". To return the bit data from "0" to "1", erase flash memory.
- All commands are ignored during automatic programming.
- If a hardware reset occurs during programming, the data being programmed to the current address is not guaranteed. Retry from the chip-erase or sector-erase command.

### **■ Flash Memory Programming Procedure**

- Figure 27.6-1 shows the sample procedure for programming into flash memory. The hardware sequence flags can be used to check the operating state of the automatic algorithm in flash memory. The data polling flag (DQ7) is used for checking the completion of programming into flash memory in this example.
- Flag check data should be read from the address where data was last written.
- Because the data polling flag (DQ7) and execution time-out flag (DQ5) are updated at the same time, the data polling flag (DQ7) must be checked even when the execution time-out flag (DQ5) is "1".
- Similarly, the toggle bit flag (DQ6) must be checked as it stops toggling at the same time as when the execution time-out flag (DQ5) changes to "1".

Figure 27.6-1 Sample Procedure for Programming into Flash Memory



### **27.6.3 Erasing All Data from Flash Memory (Chip Erase)**

---

**This section describes the procedure for issuing the chip erase command to erase all data from flash memory.**

---

#### **■ Erasing Data from Flash Memory (Chip Erase)**

- To erase all data from flash memory, send the chip erase command in the command sequence table continuously from the CPU to flash memory.
- The chip erase command is executed in six bus operations. Chip erasing is started upon completion of the sixth programming cycle.
- Before chip erasing, the user need not perform programming into flash memory. During execution of the automatic erase algorithm, flash memory automatically programs "0" before erasing all cells automatically.

#### **■ Notes on Chip Erase**

- The chip erase command is accepted only when all sectors have been program-enabled. The chip erase command is ignored if the bit for any sector in the flash memory sector write control registers (SWRE0/SWRE1) has been set to "0" (to program-disable or accidental write protect the sector).
- If a hardware reset occurs during erasure, the data being erased from flash memory is not guaranteed.

## 27.6.4 Erasing Arbitrary Data from Flash Memory (Sector Erasing)

---

This section explains the procedure for entering the sector erase command to erase any sector in flash memory. Sector-by-sector erasing is enabled and multiple sectors can be specified at a time.

---

### ■ Erasing Arbitrary Data from Flash Memory (Sector Erase)

To erase data from an arbitrary sector in flash memory, send the sector erase command in the command sequence table continuously from the CPU to flash memory.

#### ● Specifying a sector

- The sector erase command is executed in six bus operations. A minimum of 50  $\mu$ s sector erase wait is started by specifying the address for the sixth cycle as the address in the target sector and writing the sector erase code (30<sub>H</sub>) as data.
- To erase data from more than one sector, program the erase code (30<sub>H</sub>) to the sector address to be erased, following the above.

#### ● Notes on specifying two or more sectors

- Sector erasing is started upon completion for a minimum of 50  $\mu$ s sector erase wait period after the last sector erase code has been programmed.
- To erase data from two or more sectors simultaneously, input the erase sector addresses and the erase code (the sixth cycle of the command sequence) within 50  $\mu$ s. If the erase code is input after 50  $\mu$ s or over, it cannot be accepted upon completion of the sector erase wait period.
- The sector erase timer flag (DQ3) can be used to check whether it is valid to write consecutive sector erase codes.
- When reading the sector erase timer flag (DQ3), specify the address of the sector to be erased.

### ■ Flash Memory Sector Erasing Procedure

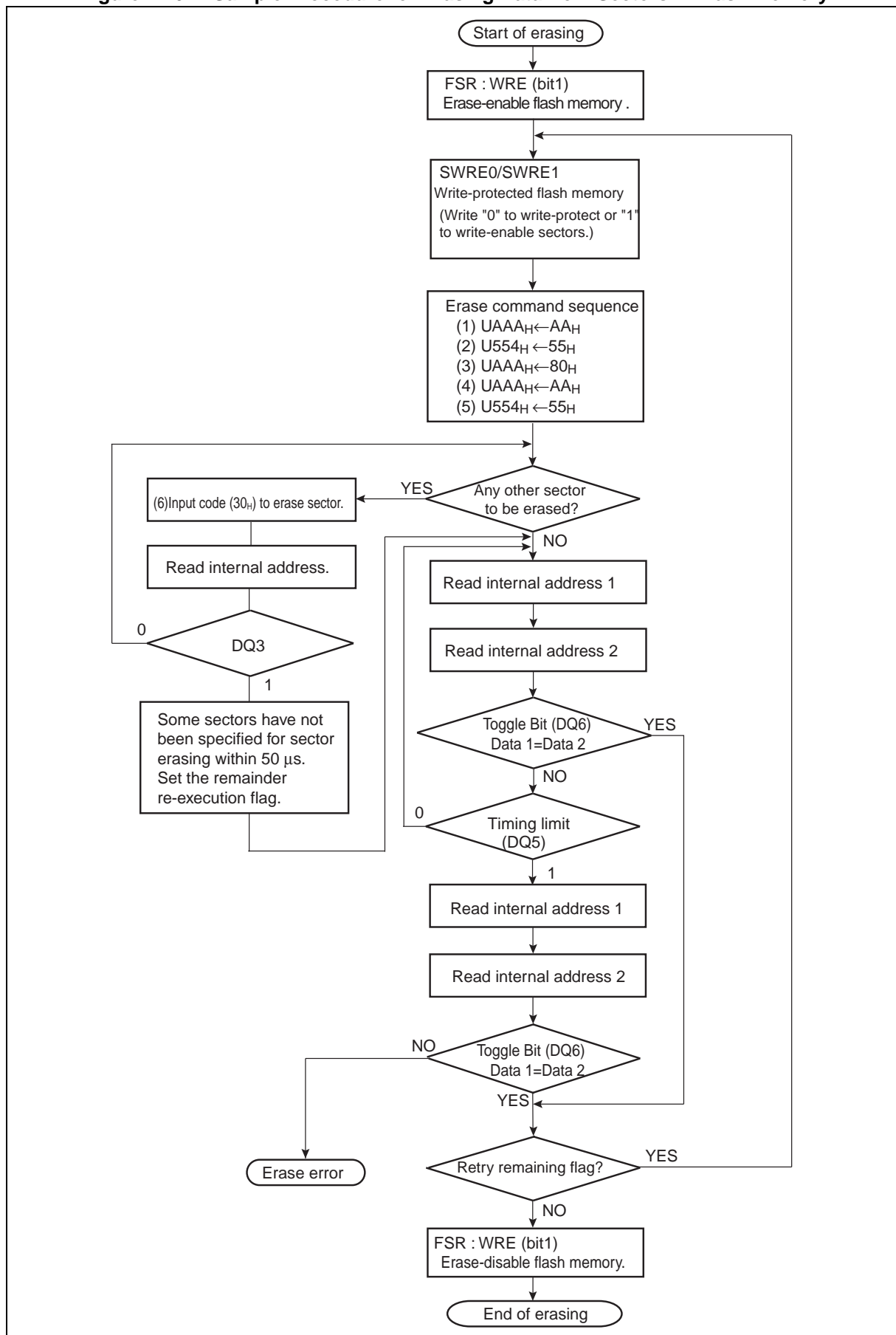
- The hardware sequence flags can be used to check the operating state of the automatic algorithm in flash memory. Figure 27.6-2 shows a sample of flash memory sector erasing procedure. In this example, the toggle bit flag (DQ6) is used to check that sector erasing is completed.
- The toggle bit flag (DQ6) stops toggling the output concurrently with the change of the execution time-out flag (DQ5) to "1". So the toggle bit flag (DQ6) must be checked even when the execution time-out flag (DQ5) is "1".
- Similarly, the data polling flag (DQ7) changes concurrently with the transition of the execution time-out flag (DQ5), so the data polling flag (DQ7) must be checked.

### ■ Notes on Erasing Data from Sectors

If a hardware reset occurs during erasing data from a sector, the data being erased is not guaranteed. Retry erasing the same sector.



Figure 27.6-2 Sample Procedure for Erasing Data from Sectors in Flash Memory



## 27.6.5 Suspending Sector Erasing from Flash Memory

---

This section explains the procedure for entering the sector erase suspend command to suspend sector erasing from flash memory. Data can be read from sectors not being erased.

---

### ■ Suspending Sector Erasing from Flash Memory

- To suspend flash memory sector erasing, send the sector erase suspend command in the command sequence table from the CPU to flash memory.
- The sector erase suspend command suspends the sector erase currently being performed, allowing data to be read from sectors that are currently not being erased.
- The sector erase suspend command is only enabled during the sector erase period including the erase wait time; it is ignored during the chip erase period or during programming.
- The sector erase suspend command is executed when the erase suspend code (B0<sub>H</sub>) is programmed. At this time, specify an arbitrary address in the sector specified for erasure. If an attempt is made to execute the sector erase suspend command again when sector erasing has been suspended, the command input again is ignored.
- When the sector erase suspend command is input during the sector erase wait period, the sector erase wait state ends immediately, the erasing is interrupted, and the erase stop state occurs.
- When the erase suspend command is input during sector erasing after the sector erase wait period, the erase suspend state occurs after a maximum of 20μs.

### ■ Note

Before issuing a suspend command, wait for 20 ms after issuing the sector erase command or sector erase resume command.

## **27.6.6 Resuming Sector Erasing from Flash Memory**

---

**This section explains the procedure for entering the sector erase resume command to resume suspended erasing of a sector in flash memory.**

---

### **■ Resuming Sector Erasing from Flash Memory**

- To resume suspended sector erasing, send the sector erase resume command in the command sequence table from the CPU to flash memory.
- The sector erase resume command resumes sector erasing suspended by the sector erase suspend command. The sector erase resume command is executed by writing erase resume code (30<sub>H</sub>). At this time, specify an arbitrary address in the sector specified for erasure.
- The sector erase resume command input during sector erasing is ignored.

## 27.7 Operation of Dual-Operation Flash Memory

The following are special notes for when using dual-operation flash memory.

- Interrupt occurrence when rewriting upper bank
- Setup Procedure of Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN)

### ■ An interrupt occurs when rewriting the upper bank.

Dual-operation flash memory is configured with two banks, but it is not possible to execute the erase/programming and read for the same bank like old flash memory.

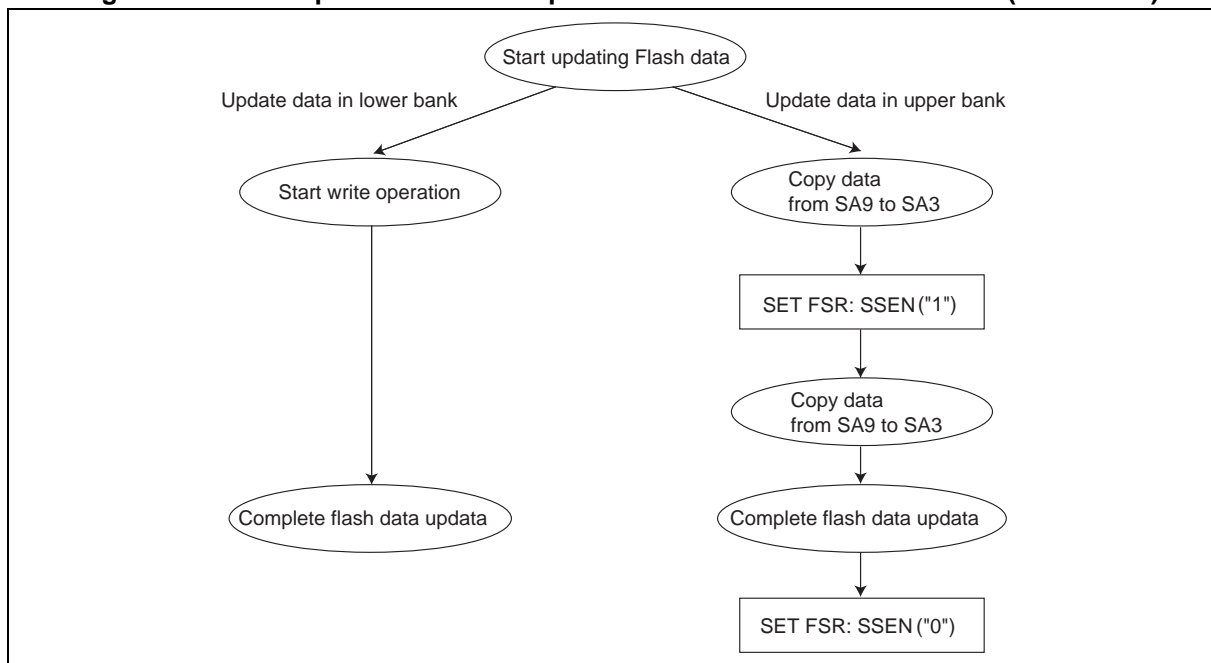
SA9 includes an interrupt vector, so the interrupt vector cannot be read correctly from the CPU when an interrupt occurs during upper bank programming. When programming the upper bank, the sector conversion enable bit needs to be set to "1" (FSR:SSEN=1). Therefore, the same data needs to be copied to SA3 and SA9 before setting the sector conversion enable bit (FSR:SSEN) in order to read the interrupt vector data from SA3 for when an interrupt occurs.

### ■ Setup Procedure of Sector Conversion Enable Bit (FSR:SSEN)

Figure 27.7-1 shows the setup procedure for the sector conversion enable bit (FSR:SSEN).

When programming data of the upper bank, the FSR:SSEN bit needs to be set to "1". Also, during programming to the flash memory, the sector conversion enable bit (FSR:SSEN) setting cannot be changed. The sector conversion enable bit (FSR:SSEN) must be set either before or after programming to the flash memory. Interrupt enable is prohibited when the FSR:SSEN bit is set. Interrupt needs to be enabled after the sector conversion enable bit (FSR:SSEN) is set.

**Figure 27.7-1 Setup Procedure Example of Sector Conversion Enable Bit (FSR:SSEN)**



### ■ Programming/Erasing Operation

When an interrupt occurs during programming/erasing to the flash memory, programming to the flash memory is prohibited within the interrupt routine.

When multiple programming/erasing routines exist, execute a programming/erasing routine after the current programming/erasing routine is finished.

State transition from programming/erasing mode (clock mode and standby mode) during programming/erasing to the flash memory is prohibited. Execute state transition after programming/erasing is finished.

**MB95110B/M Series****27.8 Flash Security**

---

**The flash security controller function prevents the contents of flash memory from being read through external pins.**

---

**■ Flash Security**

Writing protection code "01<sub>H</sub>" to a flash memory address (4000<sub>H</sub>) restricts access to flash memory, barring read/write access to flash memory from any external pin. Once flash memory has been protected, the function cannot be unlocked until the chip erase command is executed.

Note that only addresses 5554<sub>H</sub> and AAAA<sub>H</sub> can be read as exceptions.

It is advisable to code the protection code at the end of flash programming. This is to avoid unnecessary protection during programming.

Once flash memory has been protected, the chip erase operation is required before it can be reprogrammed.

## **27.9 Notes on Using Dual-Operation Flash Memory**

---

**This section describes points to note when using dual-operation flash memory.**

---

### **■ Restrictions on Toggle Bit Flag (DQ6)**

When using dual-operation flash memory (flash memory write control program is executed from flash memory),

the toggle bit flag (DQ6) cannot be used to check the status during writing or erasing.

Accordingly,

please use the data polling flag (DQ7) to check the internal status of flash memory after writing to the flash memory or erasing a sector as shown in the examples in Figure 27.6-1 and Figure 27.6-2.

Note that this precaution does not apply if running the flash write control program from RAM.

### **■ Notes on F<sup>2</sup>MC-8FX Software Development Support Environment (MB95FV100D and MB2146-09)**

- Writing or erasing the lower bank (1000<sub>H</sub> to 3FFF<sub>H</sub>) is not possible.
- Do not execute chip erasing.

# **CHAPTER 28**

---

## **256-Kbit**

# **FLASH MEMORY**

**This chapter describes the functions and operations of 256-Kbit flash memory.**

- 28.1 Overview of 256-Kbit Flash Memory
- 28.2 Sector Configuration of Flash Memory
- 28.3 Register of Flash Memory
- 28.4 Starting the Flash Memory Automatic Algorithm
- 28.5 Checking the Automatic Algorithm Execution Status
- 28.6 Flash Memory Program/Erase
- 28.7 Flash Security



## **28.1 Overview of 256-Kbit Flash Memory**

---

**256-Kbit flash memory is located from 8000<sub>H</sub> to FFFF<sub>H</sub> on the CPU memory map. The function of the flash memory interface circuit provides read access and program access from the CPU to flash memory.**

---

### **■ Overview of 256-Kbit Flash Memory**

The following methods can be used to program (write) and erase data into/from flash memory:

- Programming/erasing using a parallel writer
- Programming/erasing using a dedicated serial writer
- Programming/erasing by program execution

As flash memory by program execution can be programmed and erased by the instructions from the CPU via the flash memory interface circuit, you can efficiently reprogram (update) program code and data in flash memory with the device mounted on a circuit board.

### **■ Features of 256-Kbit Flash Memory**

- 32K bytes × 8 bits sector configuration
- Automatic program algorithm (Embedded Algorithm)
- Detection of completion of programming/erasing using the data polling or toggle bit function
- Detection of completion of programming/erasing by CPU interrupts
- Compatible with JEDEC standard commands
- Programming/erase count (minimum): 10,000 times

### **■ Programming and Erasing Flash Memory**

- It is not possible to write to and read from flash memory at the same time.
- To program/erase data into/from flash memory, first copy the program on the flash memory to RAM, and then execute the copied program on RAM, so that writing to the flash memory can be performed.

## 28.2 Sector Configuration of Flash Memory

This section explains the sector configuration of flash memory.

### ■ Sector Configuration of 256-Kbit Flash Memory

Figure 28.2-1 shows the sector configuration of the 256-Kbit flash memory. The upper and lower addresses of each sector are given in the figure.

**Figure 28.2-1 Sector Configuration of 256-Kbit Flash Memory**

Flash memory	CPU address	Programmer address*
32K bytes	8000 <sub>H</sub>	18000 <sub>H</sub>
	FFFF <sub>H</sub>	1FFFF <sub>H</sub>

\*: The programmer address is equivalent to the CPU address, which is used when data is written to the flash memory using parallel programmer.  
When a parallel programmer is used for programming/erasing, the programmer address is used for programming/erasing.

## 28.3 Register of Flash Memory

This section shows the register of the flash memory.

### ■ Register of the Flash Memory

Figure 28.3-1 Register of the Flash Memory

Flash Memory Status Register (FSR)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0072 <sub>H</sub>	—	—	RDYIRQ	RDY	Reserved	IRQEN	WRE	Reserved	000X0000 <sub>B</sub>
	R0/WX	R0/WX	R(RM1),W	R/WX	R/W0	R/W	R/W	R/W0	

R/W : Readable/writable (Read value is the same as write value)  
 R(RM1), W : Readable/writable (Read value is different from write value, "1" is read by read-modify-write (RMW) instruction)  
 R/WX : Read only (Readable, writing has no effect on operation)  
 R/W0 : Reserved bit (Write value is "0", read value is the same as write value)  
 R0/WX : Undefined bit (Read value is "0", writing has no effect on operation)  
 X : Indeterminate

## MB95110B/M Series

### 28.3.1 Flash Memory Status Register (FSR)

Figure 28.3-2 lists the functions of the flash memory status register (FSR).

#### ■ Flash Memory Status Register (FSR)

Figure 28.3-2 Flash Memory Status Register (FSR)

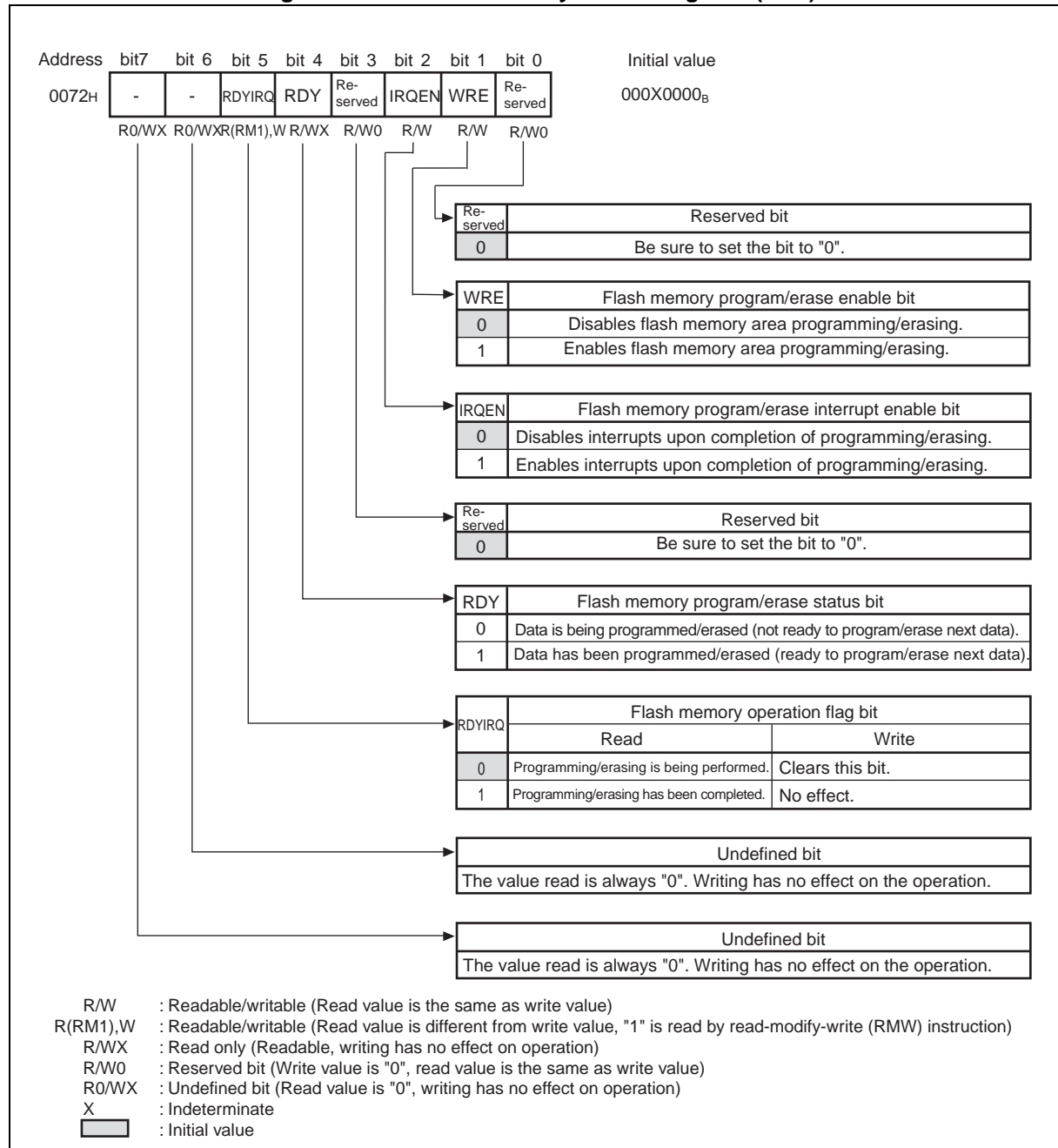


Table 28.3-1 Functions of Flash Memory Status Register (FSR)

Bit name		Function
bit7, bit6	-: Undefined bits	The value read is always "0". Writing has no effect on the operation.
bit5	RDYIRQ: Flash memory operation flag bit	<p>This bit shows the operating state of flash memory. The RDYIRQ bit is set to "1" upon completion of the flash memory automatic algorithm when flash memory programming/erasing is completed.</p> <ul style="list-style-type: none"> <li>• An interrupt request occurs when the RDYIRQ bit is set to "1" if interrupts triggered by the completion of flash memory programming/erasing have been enabled (FSR: IRQEN=1).</li> <li>• If the RDYIRQ bit is set to "0" when flash memory programming/erasing is completed, further flash memory programming/erasing is disabled.</li> </ul> <p><b>Setting the bit to "0"</b>: Clears the bit.  <b>Setting the bit to "1"</b>: Has no effect on the operation.  "1" is read from the bit whenever a read-modify-write (RMW) instruction is used.</p>
bit4	RDY: Flash memory program/erase status bit	<p>This bit shows the programming/erasing status of flash memory.</p> <ul style="list-style-type: none"> <li>• Flash memory programming/erasing cannot be performed with the RDY bit set to "0".</li> <li>• A read/reset command can be accepted even when the RDY bit contains "0". The RDY bit is set to "1" upon completion of programming/erasing.</li> <li>• It takes a delay of two machine clock (MCLK) cycles after the issuance of a program/erase command for the RDY bit to be set to "0". Read this bit after, for example, inserting NOP twice after issuing the program/erase command.</li> </ul>
bit3	Reserved: Reserved bit	Be sure to set this bit to "0".
bit2	IRQEN: Flash memory program/erase interrupt enable bit	<p>This bit enables or disables the generation of interrupt requests in response to the completion of flash memory programming/erasing.</p> <p><b>Setting the bit to "0"</b>: Prevents an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1).</p> <p><b>Setting the bit to "1"</b>: Causes an interrupt request from occurring even when the flash memory operation flag bit is set to "1" (FSR: RDYIRQ=1).</p>
bit1	WRE: Flash memory program/erase enable bit	<p>This bit enables or disables the programming/erasing of data into/from the flash memory area. Set the WRE bit before invoking a flash memory program/erase command.</p> <p><b>Setting the bit to "0"</b>: Prevents a program/erase signal from being generated even when a program/erase command is input.</p> <p><b>Setting the bit to "1"</b>: Allows flash memory programming/erasing to be performed after a program/erase command is input.</p> <p>When flash memory is not to be programmed or erased, set the WRE bit to "0" to prevent it from being accidentally programmed or erased.</p>
bit0	Reserved: Reserved bit	Be sure to set this bit to "0".

## 28.4 Starting the Flash Memory Automatic Algorithm

There are three types of commands that invoke the flash memory automatic algorithm: read/reset, write (program), and chip-erase.

### ■ Command Sequence Table

Table 28.4-1 shows commands used for programming/erasing data on flash memory.

**Table 28.4-1 Command Sequence Table**

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	Address	Data
Read/ reset*	1	F <sub>X</sub> XX <sub>H</sub>	F0 <sub>H</sub>	-	-	-	-	-	-	-	-	-	-
	4	UAAA <sub>H</sub>	AA <sub>H</sub>	U554 <sub>H</sub>	55 <sub>H</sub>	UAAA <sub>H</sub>	F0 <sub>H</sub>	RA	RD	-	-	-	-
Programming	4	UAAA <sub>H</sub>	AA <sub>H</sub>	U554 <sub>H</sub>	55 <sub>H</sub>	UAAA <sub>H</sub>	A0 <sub>H</sub>	PA	PD	-	-	-	-
Chip erasing	6	XAAA <sub>H</sub>	AA <sub>H</sub>	X554 <sub>H</sub>	55 <sub>H</sub>	XAAA <sub>H</sub>	80 <sub>H</sub>	XAAA <sub>H</sub>	AA <sub>H</sub>	X554 <sub>H</sub>	55 <sub>H</sub>	XAAA <sub>H</sub>	10 <sub>H</sub>

- RA : Read address
- PA : Write (program) address
- RD : Read data
- PD : Write (program) data
- U : Upper 4 bits same as RA and PA
- F<sub>X</sub> : FF/FE
- X : Arbitrary address

\*: Both of the two types of read/reset command can reset the flash memory to read mode.

#### Notes:

- Addresses in the table are the values in the CPU memory map. All addresses and data are hexadecimal values. However, "X" is an arbitrary value.
- Address "U" in the table is not arbitrary, whose four bits (bit15 to bit12) must have the same value as RA and PA.

Example: If RA = C48E<sub>H</sub>, U = C; If PA = 1024<sub>H</sub>, U=1

### ■ Notes on Issuing Commands

Pay attention to the following points when issuing commands in the command sequence table:

The upper address U bits (bit15 to bit12) used when commands are issued must have the same value as RA and PA, from the first command on.

If the above measures are not followed, commands are not recognized normally. Execute a reset to initialize the command sequencer in the flash memory.

## 28.5 Checking the Automatic Algorithm Execution Status

As the flash memory uses the automatic algorithm for a process flow for programming/erasing, you can check its internal operating status with hardware sequence flags.

### ■ Hardware Sequence Flag

#### ● Overview of hardware sequence flag

The hardware sequence flag consists of the following 3-bit outputs:

- Data Polling Flag (DQ7)
- Toggle Bit Flag (DQ6)
- Execution Time-out Flag (DQ5)

The hardware sequence flags tell whether the write (program) or chip-erase command has been terminated and whether an erase code write can be performed.

You can reference hardware sequence flags by read access to the address of each relevant sector in flash memory after setting a command sequence.

Table 28.5-1 shows the bit allocation of the hardware sequence flags.

**Table 28.5-1 Bit Allocation of Hardware Sequence Flags**

Bit No.	7	6	5	4	3	2	1	0
Hardware sequence flag	DQ7	DQ6	DQ5	–	–	–	–	–

- To know whether the automatic write, chip-erase, command is being executed or has been terminated, check the hardware sequence flags or the flash memory program/erase status bit in the flash memory status register (FSR:RDY). After programming/erasing is terminated, flash memory returns to the read/reset state.
- When creating a write/erase program, read data after checking the termination of automatic writing/erasing with the DQ7, DQ6, and DQ5 flags.

#### ● Explanation of hardware sequence flag

Table 28.5-2 lists the functions of the hardware sequence flag.

**Table 28.5-2 List of Hardware Sequence Flag Functions**

State		DQ7	DQ6	DQ5
State transition during normal operation	Programming → Programming completed (when write address has been specified)	$\overline{\text{DQ7}} \rightarrow \text{DATA: 7}$	Toggle → DATA: 6	0 → DATA: 5
	Chip erasing → Erasing completed	0 → 1	Toggle → Stop	0 → 1
Abnormal operation	Write	$\overline{\text{DQ7}}$	Toggle	1
	Chip erase	0	Toggle	1

## 28.5.1 Data Polling Flag (DQ7)

The data polling flag (DQ7) is a hardware sequence flag used to indicate that the automatic algorithm is being executing or has been completed using the data polling function.

### ■ Data Polling Flag (DQ7)

Table 28.5-3 and Table 28.5-4 show the state transition of the data polling flag.

**Table 28.5-3 State Transition of Data Polling Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ7	$\overline{\text{DQ7}} \rightarrow \text{DATA: 7}$	$0 \rightarrow 1$

**Table 28.5-4 State Transition of Data Polling Flag (During Abnormal Operation)**

Operating state	Programming	Chip erase
DQ7	$\overline{\text{DQ7}}$	0

#### ● At programming

When read access takes place during execution of the automatic write algorithm, the flash memory outputs the inverted value of bit7 in the last data written to DQ7.

If read access takes place on completion of the automatic write algorithm, the flash memory outputs bit7 of the value read from the read-accessed address to DQ7.

#### ● At chip erasing

When read access is made to the sector currently being erased during execution of the chip erase automatic algorithm, bit7 of flash memory outputs "0". Bit7 of flash memory outputs "1" upon completion of chip erasing.

#### Note:

Once the automatic algorithm has been started, read access to the specified address is ignored. Data reading is allowed after the data polling flag (DQ7) is set to "1". Data reading after the end of the automatic algorithm should be performed following read access made to confirm the completion of data polling.



## 28.5.2 Toggle Bit Flag (DQ6)

The toggle bit flag (DQ6) is a hardware sequence flag used to indicate that the automatic algorithm is being executed or has been completed using the toggle bit function.

### ■ Toggle Bit Flag (DQ6)

Table 28.5-5 and Table 28.5-6 show the state transition of the toggle bit flag.

**Table 28.5-5 State Transition of Toggle Bit Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ6	Toggle → DATA: 6	Toggle → Stop

**Table 28.5-6 State Transition of Toggle Bit Flag (During Abnormal Operation)**

Operating state	Programming	Chip erase
DQ6	Toggle	Toggle

#### ● At programming and chip erasing

- When read access is made continuously during execution of the automatic write algorithm or chip-erase/sector-erase algorithm, the flash memory toggles the output between "1" and "0" at each read access.
- When read access is made continuously after the automatic write algorithm or chip-erase/sector-erase algorithm is terminated, the flash memory outputs bit6 (DATA:6) of the value read from the read address at each read access.

### 28.5.3 Execution Time-out Flag (DQ5)

The execution time-out flag (DQ5) is a hardware sequence flag indicating that the automatic algorithm has been executed beyond the specified time (required for programming/erasing) internal to the flash memory.

#### ■ Execution Time-out Flag (DQ5)

Table 28.5-7 and Table 28.5-8 show the state transition of the execution time-out flag.

**Table 28.5-7 State Transition of Execution Time-out Flag (During Normal Operation)**

Operating state	Programming → Programming completed	Chip erase → Erasing completed
DQ5	0 → DATA: 5	0 → 1

**Table 28.5-8 State Transition of Execution Time-out Flag (During Abnormal Operation)**

Operating state	Programming	Chip erase
DQ5	1	1

#### ● At programming and chip erasing

When read access is made with the write or chip-erase automatic algorithm invoked, the flag outputs "0" when the algorithm execution time is within the specified time (required for programming/erasing) or "1" when it exceeds that time.

The execution time-out flag (DQ5) can be used to check whether programming/erasing has succeeded or failed regardless of whether the automatic algorithm has been running or terminated. When the execution time-out flag (DQ5) outputs "1", it indicates that programming has failed if the automatic algorithm is still running for the data polling or toggle bit function.

If an attempt is made to write "1" to a flash memory address holding "0", for example, the flash memory is locked, preventing the automatic algorithm from being terminated and valid data from being output from the data polling flag (DQ7). As the toggle bit flag (DQ6) does not stop toggling, the time limit is exceeded and the execution time-out flag (DQ5) outputs "1". The state in which the execution time-out flag (DQ5) outputs "1" means that the flash memory has not been used correctly; it does not mean that the flash memory is defective. When this state occurs, execute the reset command.

## **28.6 Flash Memory Program/Erase**

---

**This section describes the individual procedures for flash memory reading/resetting, programming, and chip-erasing by entering their respective commands to invoke the automatic algorithm.**

---

### **■ Details of Programming/Erasing Flash Memory**

The automatic algorithm can be invoked by writing the read/reset, program, and chip-erase command sequence to flash memory from the CPU. Writing command sequence to flash memory from the CPU must always be performed continuously. The termination of the automatic algorithm can be checked by the data polling function. After the automatic algorithm terminates normally, the flash memory returns to the read/reset state.

The individual operations are explained in the following order:

- Enter read/reset state.
- Program data.
- Erase all data (chip-erase).

## **28.6.1 Placing Flash Memory in the Read/Reset State**

---

**This section explains the procedure for entering the read/reset command to place flash memory in the read/reset state.**

---

### **■ Placing Flash Memory in the Read/Reset State**

- To place flash memory in the read/reset state, send the read/reset command in the command sequence table continuously from the CPU to flash memory.
- The read/reset command is available in two different command sequences: one involves a single bus operation and the other involves four bus operations, which are essentially the same.
- Since the read/reset state is the initial state of flash memory, the flash memory always enters this state after the power is turned on and at the normal termination of a command. The read/reset state is also described as the wait state for command input.
- In the read/reset state, read access to flash memory enables data to be read. As is the case with masked ROM, program access from the CPU can be made.
- Read access to flash memory does not require the read/reset command. If a command is not terminated normally, use the read/reset command to initialize the automatic algorithm.

## **28.6.2 Programming Data into Flash Memory**

---

**This section explains the procedure for entering the write (program) command to program data into flash memory.**

---

### **■ Programming Data into Flash Memory**

- To start the automatic algorithm for programming data into flash memory, send the program command in the command sequence table continuously from the CPU to flash memory.
- Upon completion of data programming to a target address in the fourth cycle, the automatic algorithm starts automatic programming.

#### **● How to specify addresses**

Programming (writing) can be performed even in any order of addresses or across a sector boundary. Data written by a single program command is only one byte.

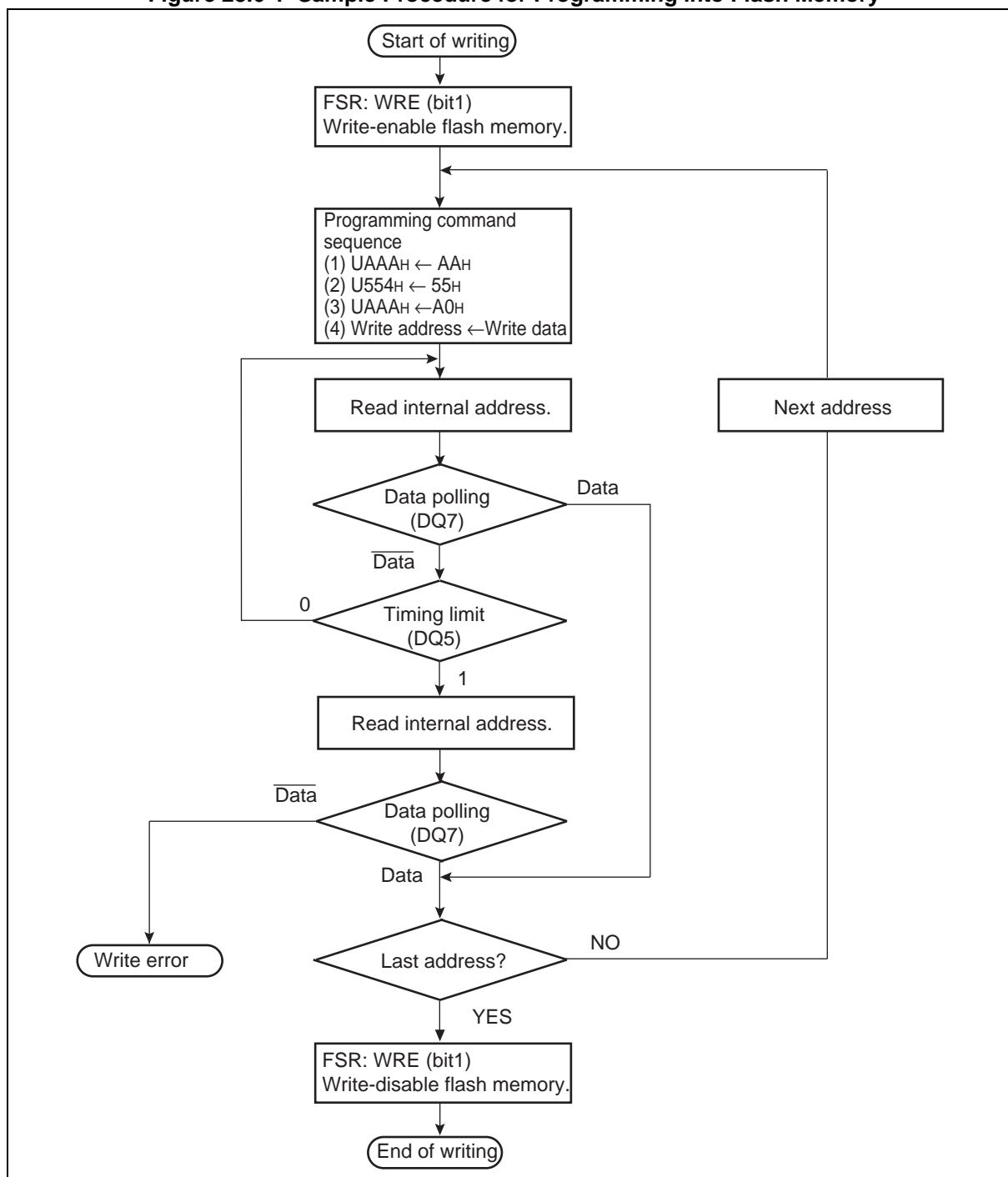
#### **● Notes on programming data**

- Bit data cannot be returned from "0" to "1" by programming. When bit data "1" is programmed to bit data "0", the data polling function (DQ7) or toggle operation (DQ6) is not terminated, the flash memory element is determined to be defective, and the execution time-out flag (DQ5) detects an error to indicate that the specified programming time has been exceeded. When data is read in the read/reset state, the bit data remains "0". To return the bit data from "0" to "1", erase flash memory.
- All commands are ignored during automatic programming.
- If a hardware reset occurs during programming, the data being programmed to the current address is not guaranteed. Retry from the chip-erase command.

### **■ Flash Memory Programming Procedure**

- Figure 28.6-1 shows the sample procedure for programming into flash memory. The hardware sequence flags can be used to check the operating state of the automatic algorithm in flash memory. The data polling flag (DQ7) is used for checking the completion of programming into flash memory in this example.
- Flag check data should be read from the address where data was last written.
- Because the data polling flag (DQ7) and execution time-out flag (DQ5) are updated at the same time, the data polling flag (DQ7) must be checked even when the execution time-out flag (DQ5) is "1".
- Similarly, the toggle bit flag (DQ6) must be checked as it stops toggling at the same time as when the execution time-out flag (DQ5) changes to "1".

Figure 28.6-1 Sample Procedure for Programming into Flash Memory



### **28.6.3 Erasing All Data from Flash Memory (Chip Erase)**

---

**This section describes the procedure for issuing the chip erase command to erase all data from flash memory.**

---

#### **■ Erasing Data from Flash Memory (Chip Erase)**

- To erase all data from flash memory, send the chip erase command in the command sequence table continuously from the CPU to flash memory.
- The chip erase command is executed in six bus operations. Chip erasing is started upon completion of the sixth programming cycle.
- Before chip erasing, the user need not perform programming into flash memory. During execution of the automatic erase algorithm, flash memory automatically programs "0" before erasing all cells automatically.

#### **■ Notes on Chip Erase**

If a hardware reset occurs during erasure, the data being erased from flash memory is not guaranteed.

**MB95110B/M Series****28.7 Flash Security**

---

**The flash security controller function prevents the contents of flash memory from being read through external pins.**

---

**■ Flash Security**

Writing protection code "01<sub>H</sub>" to a flash memory address (8000<sub>H</sub>) restricts access to flash memory, barring read/write access to flash memory from any external pin. Once flash memory has been protected, the function cannot be unlocked until the chip erase command is executed.

Note that only addresses 5554<sub>H</sub> and 2AAA<sub>H</sub> can be read as exceptions.

It is advisable to code the protection code at the end of flash programming. This is to avoid unnecessary protection during programming.

Once flash memory has been protected, the chip erase operation is required before it can be reprogrammed.





# ***CHAPTER 29***

---

# ***EXAMPLE OF SERIAL PROGRAMMING CONNECTION***

**This chapter describes the example of a serial programming connection.**

- 29.1 Basic Configuration of Serial Programming Connection for Flash Memory Products
- 29.2 Example of Serial Programming Connection
- 29.3 Example of Minimum Connection to Flash Microcontroller Programmer

## 29.1 Basic Configuration of Serial Programming Connection for Flash Memory Products

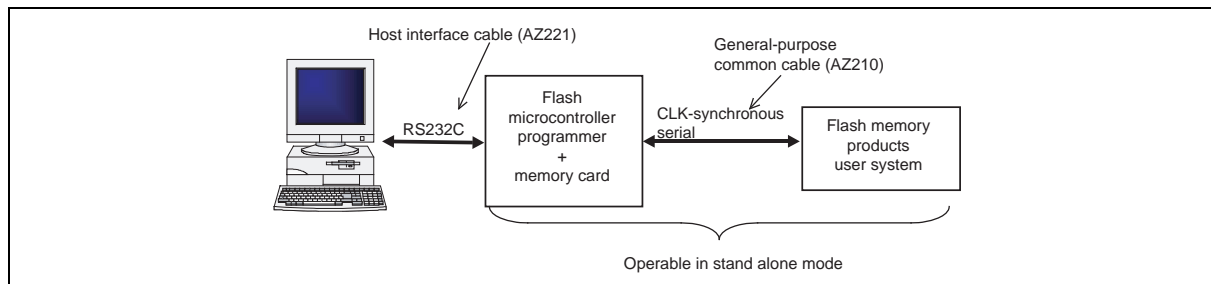
The MB95F114M(S,W)/MB95F114N(S,W)/MB95F114J(S,W)/MB95F116M(S,W)/MB95F116N(S,W)/MB95F116J(S,W)/MB95F118B(S,W)/MB95F118M(S,W)/MB95F118N(S,W)/MB95F118J(S,W)/MB95F116MA(S, W)/MB95F116NA(S, W) support flash ROM serial onboard programming (Fujitsu standard). This section describes the specifications.

### ■ Basic Configuration of Serial Programming Connection for Flash Memory Products

Flash microcontroller programmer manufactured by Yokogawa Digital Computer Co., Ltd. is used for Fujitsu standard serial onboard programming.

Figure 29.1-1 shows the basic configuration of serial programming connection for flash memory products.

**Figure 29.1-1 Basic Configuration of Serial Programming Connection for Flash Memory Products**



#### Note:

For the function and operation method of the AF220/AF210/AF120/AF110 flash microcontroller programmer and the general-purpose common cable (AZ210) for connection and connector, contact Yokogawa Digital Computer Co., Ltd.

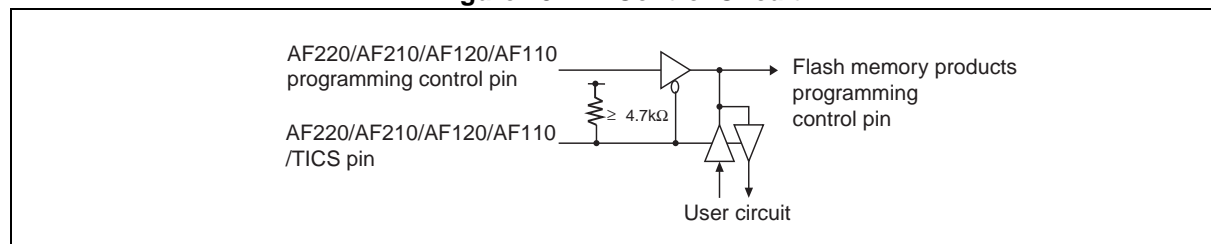
**Table 29.1-1 Pins Used for Fujitsu Standard Serial Onboard Programming**

Pin	Function	Description
MOD, P13	Mode pin	Setting MOD=High and P13=Low sets serial write mode.
X0, X1	Oscillation pins	The CPU's internal operating clock during serial write mode is the oscillator frequency divided by two. Note that a 1MHz or higher oscillator frequency must be input when performing serial writing.
$\overline{\text{RST}}$	Reset pin	-
P10/UI0	Serial data input pin	Setting P10/UI0=Low specifies that serial write mode uses clock synchronous communications. As this low input is handled by the TTXD pin of the flash microcontroller programmer, you do not need to provide a pull-down for the P10/UI0 pin.
P11/UO0	Serial data output pin	-
P12/UCK0	Serial clock input pin	Setting P12/UCK0=High sets serial write mode. As this high input is handled by the TCK pin of the flash microcontroller programmer, you do not need to provide a pull-up for the P12/UCK0 pin.
V <sub>CC</sub>	Power supply voltage supply pin	On the 3V products, the write voltage (V <sub>cc</sub> = 2.7V to 3.6V) is supplied from the user system. On the 5V products, the write voltage (V <sub>cc</sub> = 4.5V to 5.5V) is supplied from the user system.
V <sub>SS</sub>	GND pin	Common to the GND of the flash microcontroller programmer.

As the UI0, UO0, and UCK0 pins are also used by the user system, you need to provide a control circuit as shown in Figure 29.1-2 if you want to disconnect from the user circuit during serial programming.

(The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 29.1-2 for details.)

**Figure 29.1-2 Control Circuit**



● Oscillation Clock Frequency and Serial Clock Input Frequency

The permitted frequency for the input serial clock on the flash memory products is calculated from the following formula. Accordingly, modify the serial clock input frequency by setting the flash microcontroller programmer, according to the oscillation clock frequency used.

Permitted frequency for the input serial clock =  $0.125 \times$  Oscillation clock frequency

Example:

Oscillation clock frequency	Maximum serial clock frequency that can be input to the microcontroller	Maximum serial clock frequency that can be set on the AF220, AF210, AF120, and AF110	Maximum serial clock frequency that can be set on the AF200
at 4MHz	500kHz	500kHz	500kHz
at 8MHz	1MHz	850kHz	500kHz
at 10MHz	1.25MHz	1.25MHz	500kHz

**Table 29.1-2 System Configuration of the Flash Microcontroller Program (Yokogawa Digital Computer Co., Ltd.)**

Product type		Function	
Main unit	AF220/AC4P	Model with built-in Ethernet interface	/100V to 220V power adapter
	AF210/AC4P	Standard model	/100V to 220V power adapter
	AF120/AC4P	Single-key model with built-in Ethernet interface	/100V to 220V power adapter
	AF110/AC4P	Single-key model	/100V to 220V power adapter
AZ221		Writer-dedicated RS232C cable for PC/AT	
AZ210		Standard target probe (a) Length: 1 m	
FF201		Fujitsu control module for F2MC-16LX flash microcontroller	
AZ290		Remote controller	
/P2		2Mbytes PC Card (Option) Flash memory capacity: up to 128 Kbytes	
/P4		4Mbytes PC Card (Option) Flash memory capacity: up to 512 Kbytes	

Contact: Yokogawa Digital Computer Co., Ltd. Tell: +81-042-333-6222

**Note:**

Although the AF200 flash microcontroller programmer is an old model, this can be handled using the FF201 control module. The connection examples shown in the next chapter can also be used as example connections for serial writing.

## 29.2 Example of Serial Programming Connection

---

Inputting MOD="H" from TAUX3 on the AF220, AF210, AF120, or AF110 to the mode pin, which is set to MOD="L" by the user system, sets the mode to serial write mode (serial write mode: MOD="H", P12="H", P13="L").

---

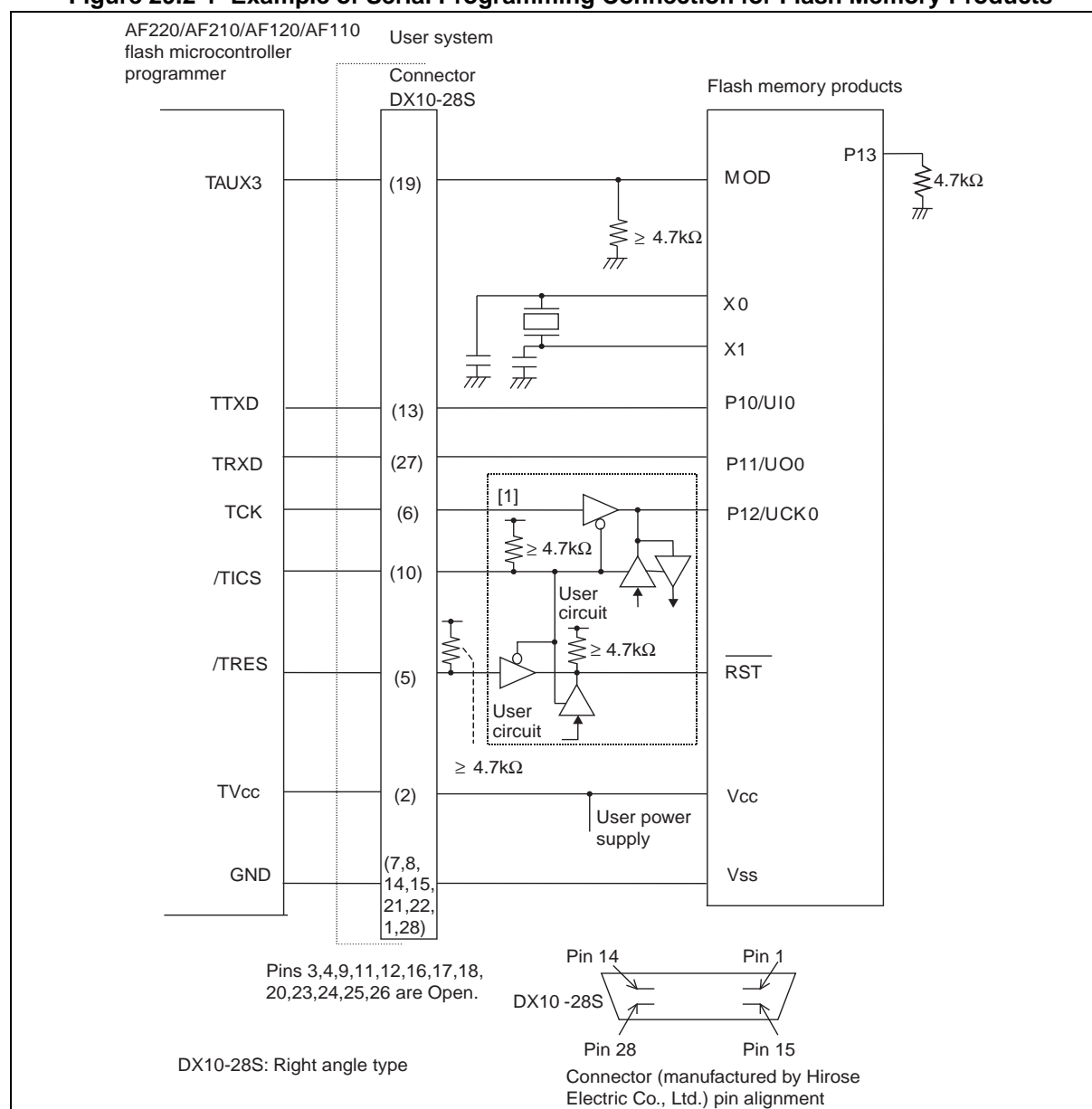
### ■ Example of Serial Programming Connection

Figure 29.2-1 shows an example connection for serial writing.

The TTXD pin on the flash microcontroller programmer is connected to P10/UI0 and outputs low until data transfer starts. Setting P10/UI0=Low in this way specifies that serial write mode uses clock synchronous communications.

Note that a user power supply is required for serial programming.

Figure 29.2-1 Example of Serial Programming Connection for Flash Memory Products

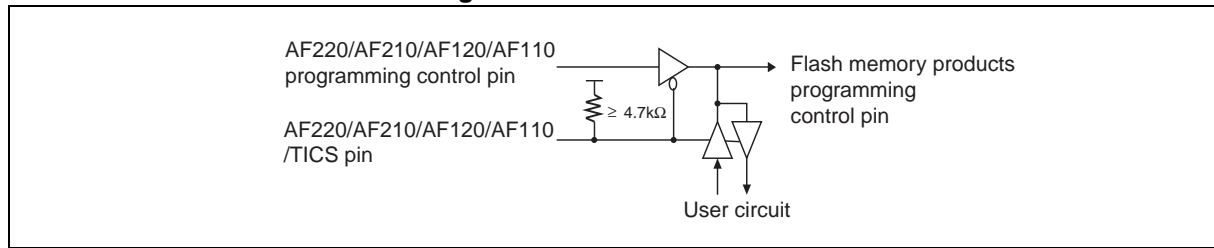


The circuit [1] shown in Figure 29.2-1 is required if you want to disconnect the UCK0 and RST pins from the user circuit during serial programming (The /TICS signal of the flash microcontroller programmer outputs low during serial writing and this disconnects the user circuit).

If it is not necessary to disconnect from the user circuit, the connection to /TICS and circuit [1] are not required. See the connection example in Figure 29.3-1.

The UI0 and UO0 pins are also used by the user system and the control circuit shown below like that used for the UCK0 pin is required if you want to disconnect from the user circuit during serial programming (The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 29.2-1 for details).

**Figure 29.2-2 Control Circuit**



Only connect to the AF220, AF210, AF120, or AF110 while the user power supply is turned off.

**Note:**

The pull-up and pull-down resistances in the above example connection are examples only and may be adjusted to suit your system. If variation in the input level to the MOD pin is possible due to noise or other factors, it is also recommended that you use a capacitor or other method to minimize noise.



## 29.3 Example of Minimum Connection to Flash Microcontroller Programmer

---

The connection between MOD and the flash microcontroller programmer is not required if the pins are set as shown in Figure 29.3-1 during serial writing (serial write mode: MOD="H", P12="H", P13="L").

---

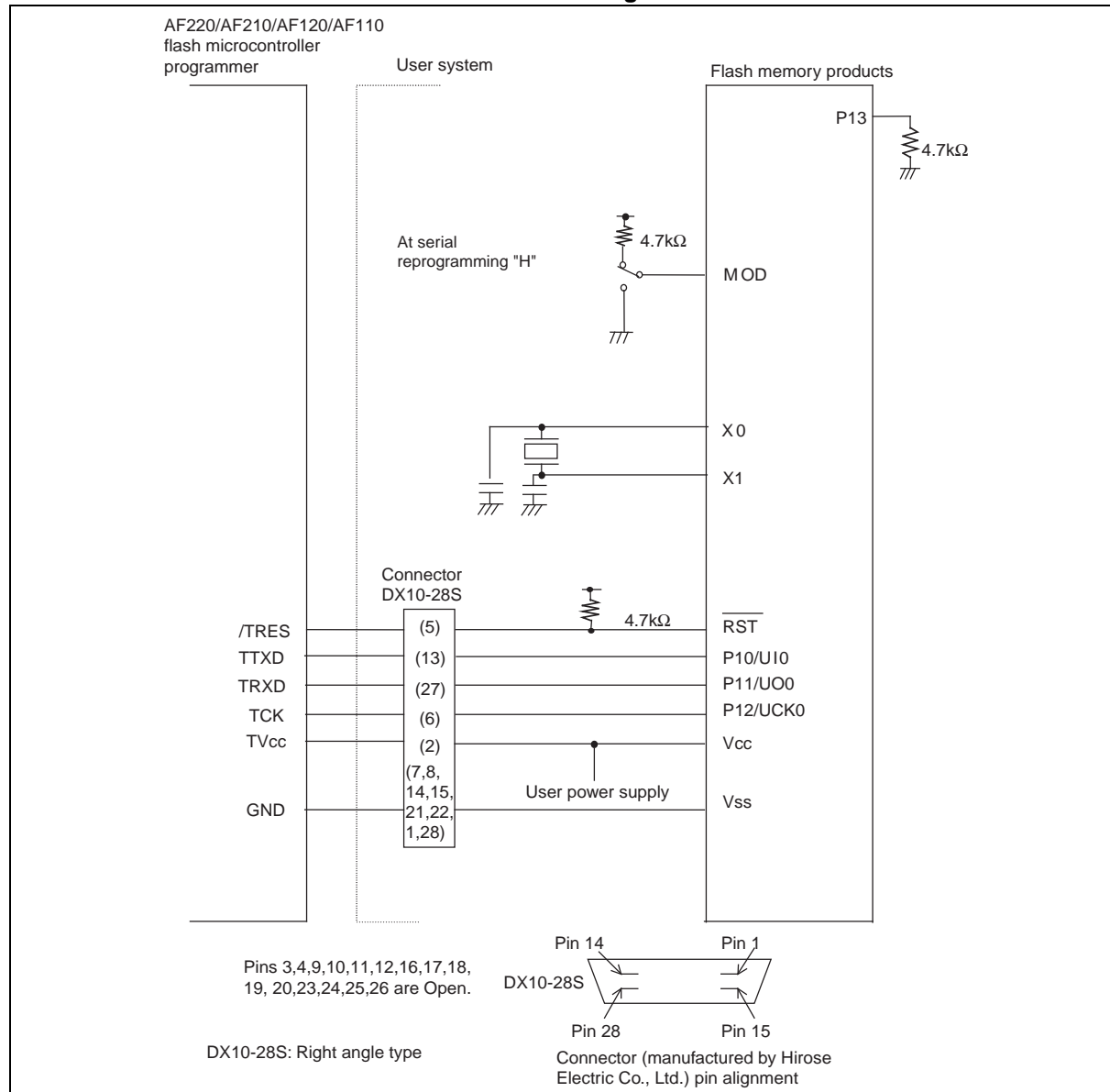
### ■ Example of Minimum Connection to Flash Microcontroller Programmer

Figure 29.3-1 shows an example of the minimum connection between the flash memory products and flash microcontroller programmer.

The TTXD pin on the flash microcontroller programmer is connected to P10/UI0 and outputs low until data transfer starts. Setting P10/UI0=Low in this way specifies that serial write mode uses clock synchronous communications.

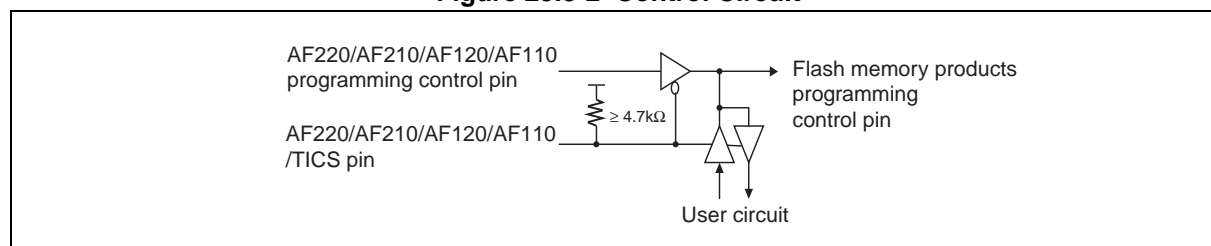
Note that a user power supply is required for serial writing.

**Figure 29.3-1 Example of Minimum Connection between Flash Memory Products and Flash Microcontroller Programmer**



As the UI0, UO0, and UCK0 pins are also used by the user system, you need to provide a control circuit as shown below if you want to disconnect from the user circuit during serial writing (The /TICS signal of the flash microcontroller programmer can be used to disconnect from the user circuit during serial writing. See the connection example in Figure 29.2-1 for details).

### Figure 29.3-2 Control Circuit



Only connect to the AF220, AF210, AF120, or AF110 while the user power supply is turned off.

**Note:**

The pull-up and pull-down resistances in the above example connection are examples only and may be adjusted to suit your system. If variation in the input level to the MOD pin is possible due to noise or other factors, it is also recommended that you use a capacitor or other method to minimize noise.

---

# **CHAPTER 30**

---

## ***DUAL OPERATION FLASH***

**This chapter (not available for MB95F116MAW/F116NAW/F116MAS/FA116NAS) describes the functions and operations of dual operation flash.**

- 30.1 Overview of Dual Operation Flash
- 30.2 Access Sector Map of Dual Operation Flash
- 30.3 Operations of Dual Operation Flash
- 30.4 Notes on Using Dual Operation Flash

## 30.1 Overview of Dual Operation Flash

---

Dual-operation flash memory consists of upper banks (16K bytes  $\times$  2 + 4K bytes  $\times$  4) and lower banks (4K bytes  $\times$  3). Unlike conventional flash products, this memory can be programmed/erased and read at the same time in banks.

This enables program execution in flash memory and write control using interrupts. In addition, a conventional process of downloading a program to be executed to RAM is not required, thereby saving the download time and eliminating the need for taking care of RAM data from power shutdown. The minimum sector is 4 K bytes long, which can be handled easily as a program/data area.

---

### ■ Features of Dual Operation Flash

Two-bank configuration, enabling simultaneous execution of an erase/program and a read

Minimum sector size of 4 K bytes contributing to easy-to-use configuration of program/data areas

The dual operation flash can use the following combinations:

Upper bank	Lower bank
Read	
Read	Program/sector erase
Program/sector erase	Read
Chip erase	

The bank on one side cannot be programmed/sector-erased while the bank on the other side is being programmed/erased.

## 30.2 Access Sector Map of Dual Operation Flash

This section describes the access sector map applicable during operation of the dual operation flash.

### ■ Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN)

During operation of the dual operation flash, upper bank SA9 and lower bank SA3 can be replaced with each other as an area containing an interrupt vector by setting the sector conversion enable bit in the flash memory status register (FSR:SSEN). Table 30.2-1 shows the relationships between FSR:SSEN and SA9/SA3 sector conversion.

**Table 30.2-1 Function of Sector Conversion Enable Bit in Flash Memory Status Register**

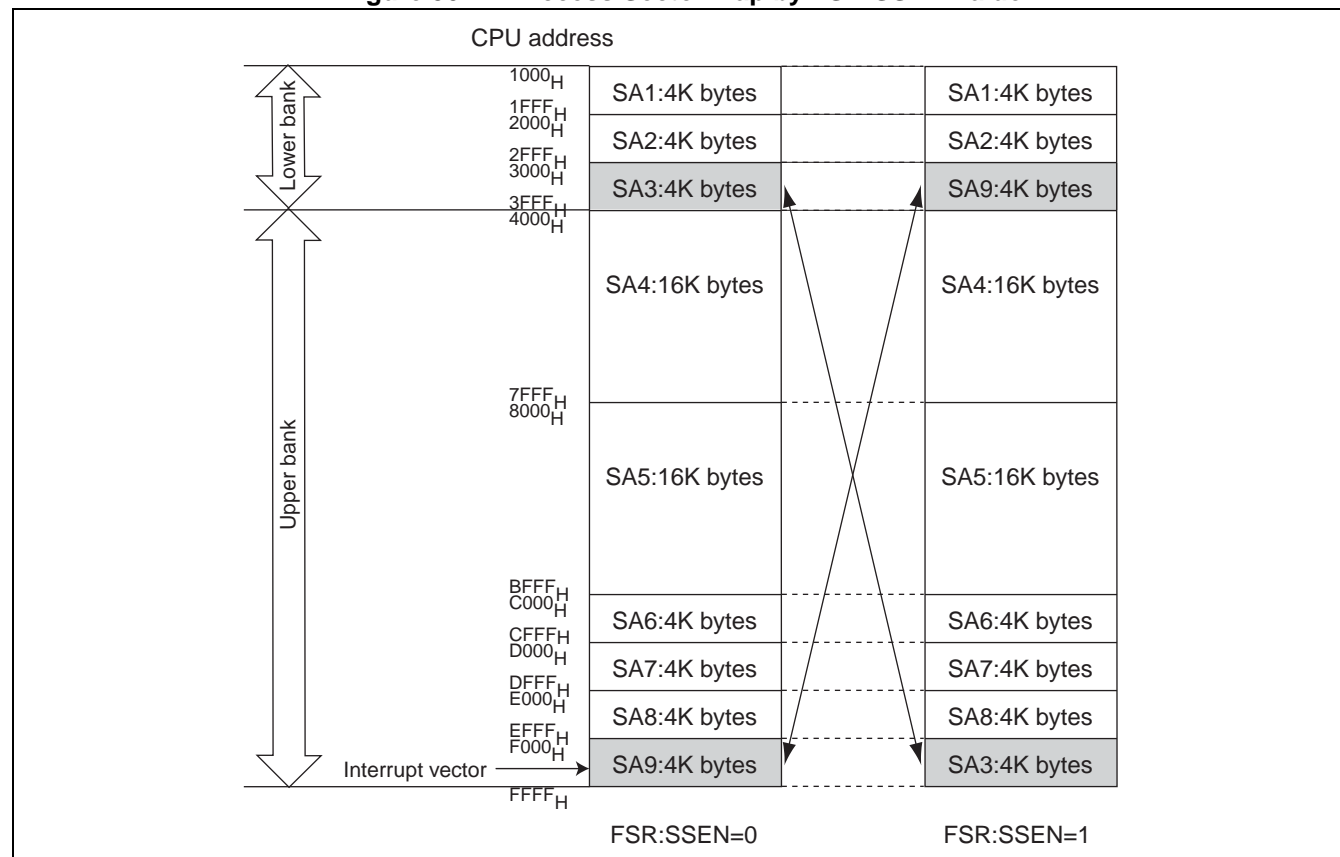
FSR:SSEN	Sector Conversion Enable Bit in Flash Memory Status Register
0 (Initial value)	SA3 and SA9 are mapped to addresses 3000 <sub>H</sub> to 3FFF <sub>H</sub> and F000 <sub>H</sub> to FFFF <sub>H</sub> , respectively. An interrupt vector exists in SA9.
1	SA9 and SA3 are mapped to addresses 3000 <sub>H</sub> to 3FFF <sub>H</sub> and F000 <sub>H</sub> to FFFF <sub>H</sub> , respectively. An interrupt vector exists in SA3.

Refer to "CHAPTER 27 DUAL-OPERATION FLASH MEMORY" for the flash memory status register (FSR).

■ **Access Sector Map Based on Sector Conversion Enable Bit (FSR:SSEN)**

Figure 30.2-1 is a flash memory access sector map based on the values of the sector conversion enable bit in the flash memory status register (FSR:SSEN).

**Figure 30.2-1 Access Sector Map by FSR:SSEN Value**



### 30.3 Operations of Dual Operation Flash

This section describes the operations of dual operation flash.

Pay attention in particular to the following points when using dual operation flash:

- Interrupt generated when upper banks are updated
- Procedure of setting the sector conversion enable bit in the flash memory status register (FSR:SSEN)

#### ■ Interrupt Generated when Upper Banks are Updated

The dual-operation flash consists of banks on two sides. Like conventional flash products, however, it cannot be erased/programmed and read at the same time in banks on the same side.

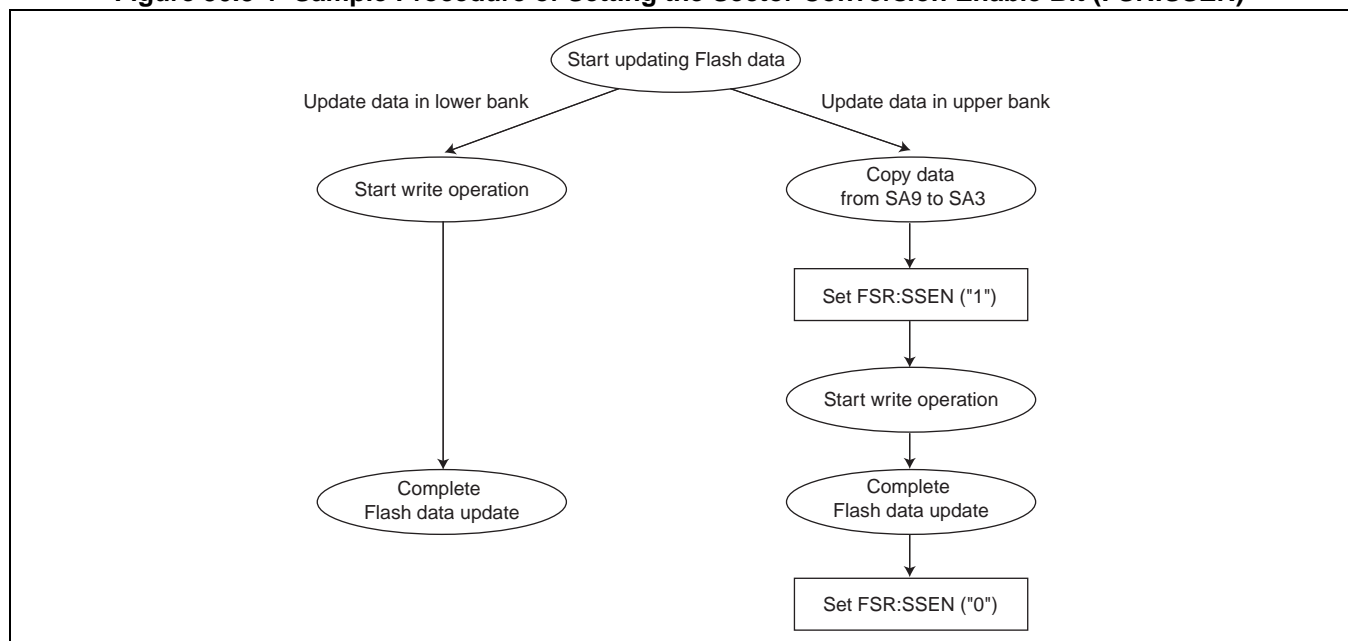
As SA9 contains an interrupt vector, an interrupt vector from the CPU cannot be read normally when an interrupt occurs during a write to an upper bank. Before an upper bank can be updated, the sector conversion enable bit must be set to "1" (FSR:SSEN = 1). When an interrupt occurs, therefore, SA3 is accessed to read interrupt vector data. The same data must be copied to SA3 and SA9 before the sector conversion enable bit (FSR:SSEN) is set.

#### ■ Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN)

Figure 30.3-1 shows a sample procedure of setting the sector conversion enable bit (FSR:SSEN).

The FSR:SSEN bit must be set to "1" before upper-bank data can be updated. Note also that it is prohibited to change the setting of the sector conversion enable bit (FSR:SSEN) during a write to flash memory. Be sure to set the sector conversion enable bit (FSR:SSEN) either before starting writing to flash memory or after completing the write. When setting the sector conversion enable bit (FSR:SSEN), disable interrupts and enable interrupts after setting the FSR:SSEN bit.

**Figure 30.3-1 Sample Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN)**





### ■ Operation During Programming/Erasing

It is prohibited to write to flash memory within an interrupt routine when an interrupt occurs during flash memory programming/erasing.

When two or more program/erase routines exist, let the interrupted program/erase routine be completed, then execute the others.

During flash memory programming/erasing, it is also prohibited to cause state transition from the current mode (clock mode or standby mode). Cause state transition after programming/erasing is completed.

## 30.4 Notes on Using Dual Operation Flash

---

This section shows the notes on using dual operation flash.

---

### ■ Notes on Software Programming Support Environment of F<sup>2</sup>MC-8FX (MB95FV100D and MB2146-09)

- Writing/erasing to/from lower bank (1000<sub>H</sub> to 3FFF<sub>H</sub>) is disabled.
- Do not erase the chip.

### ■ Restrictions on Toggle Bit Flag (DQ6)

When using dual-operation flash memory (flash memory write control program is executed from flash memory), the toggle bit flag (DQ6) cannot be used to check the status during writing or erasing. Please refer to the notes in "27.9 Notes on Using Dual-Operation Flash Memory" when writing your program.

Note that this precaution does not apply if running the flash memory write control program from RAM.

### ■ Restrictions on Toggle Bit 2 Flag (DQ2)

When using dual-operation flash memory (flash memory write control program is executed from flash memory), the toggle bit 2 flag (DQ2) cannot be used to check the status while erasing is paused.

Note that this precaution does not apply if running the flash memory write control program from RAM.



# APPENDIX

---

**This appendix explains I/O map, interrupt list, memory map, pin status, instruction overview, mask option and writing to Flash microcomputer using parallel writer.**

APPENDIX A I/O Map

APPENDIX B Table of Interrupt Causes

APPENDIX C Memory Map

APPENDIX D Pin Status of MB95110B/M series

APPENDIX E Instruction Overview

APPENDIX F Mask Option

APPENDIX G Writing to Flash Microcontroller Using Parallel Writer

## APPENDIX A I/O Map

This section explains I/O map that is used on MB95110B/M series.

### ■ I/O Map

Table A-1 MB95110B/M Series (1 / 5)

Address	Register abbreviation	Register name	R/W	Initial value
0000 <sub>H</sub>	PDR0	Port 0 data register	R/W	00000000 <sub>B</sub>
0001 <sub>H</sub>	DDR0	Port 0 direction register	R/W	00000000 <sub>B</sub>
0002 <sub>H</sub>	PDR1	Port 1 data register	R/W	00000000 <sub>B</sub>
0003 <sub>H</sub>	DDR1	Port 1 direction register	R/W	00000000 <sub>B</sub>
0004 <sub>H</sub>	(Prohibited)			
0005 <sub>H</sub>	WATR	Oscillation stabilization wait time setting register	R/W	11111111 <sub>B</sub>
0006 <sub>H</sub>	PLLC	PLL control register	R/W	00000000 <sub>B</sub>
0007 <sub>H</sub>	SYCC	System clock control register	R/W	1010x011 <sub>B</sub>
0008 <sub>H</sub>	STBC	Standby control register	R/W	00000000 <sub>B</sub>
0009 <sub>H</sub>	RSRR	Reset cause register	R	xxxxxxxx <sub>B</sub>
000A <sub>H</sub>	TBTC	Time-base timer control register	R/W	00000000 <sub>B</sub>
000B <sub>H</sub>	WPCR	Watch prescaler control register	R/W	00000000 <sub>B</sub>
000C <sub>H</sub>	WDTC	Watchdog timer control register	R/W	00000000 <sub>B</sub>
000D <sub>H</sub>	(Reserved Area)			
000E <sub>H</sub>	PDR2	Port 2 data register	R/W	00000000 <sub>B</sub>
000F <sub>H</sub>	DDR2	Port 2 direction register	R/W	00000000 <sub>B</sub>
0010 <sub>H</sub>	PDR3	Port 3 data register	R/W	00000000 <sub>B</sub>
0011 <sub>H</sub>	DDR3	Port 3 direction register	R/W	00000000 <sub>B</sub>
0012 <sub>H</sub>	(Reserved Area)			
0013 <sub>H</sub>				
0014 <sub>H</sub>	PDR5	Port 5 data register	R/W	00000000 <sub>B</sub>
0015 <sub>H</sub>	DDR5	Port 5 direction register	R/W	00000000 <sub>B</sub>
0016 <sub>H</sub>	PDR6	Port 6 data register	R/W	00000000 <sub>B</sub>
0017 <sub>H</sub>	DDR6	Port 6 direction register	R/W	00000000 <sub>B</sub>
0018 <sub>H</sub> to 0029 <sub>H</sub>	(Reserved Area)			
002A <sub>H</sub>	PDRG	Port G data register	R/W	00000000 <sub>B</sub>
002B <sub>H</sub>	DDRG	Port G direction register	R/W	00000000 <sub>B</sub>
002C <sub>H</sub>	(Reserved Area)			
002D <sub>H</sub>	PUL1	Port 1 pull-up register	R/W	00000000 <sub>B</sub>
002E <sub>H</sub>	PUL2	Port 2 pull-up register	R/W	00000000 <sub>B</sub>

**Table A-1 MB95110B/M Series (2 / 5)**

Address	Register abbreviation	Register name	R/W	Initial value
002F <sub>H</sub>	PUL3	Port 3 pull-up register	R/W	00000000 <sub>B</sub>
0030 <sub>H</sub> to 0034 <sub>H</sub>	(Reserved Area)			
0035 <sub>H</sub>	PULG	Port G pull-up register	R/W	00000000 <sub>B</sub>
0036 <sub>H</sub>	T01CR1	8/16-bit compound timer 01 control status register 1 ch.0	R/W	00000000 <sub>B</sub>
0037 <sub>H</sub>	T00CR1	8/16-bit compound timer 00 control status register 1 ch.0	R/W	00000000 <sub>B</sub>
0038 <sub>H</sub>	T11CR1	8/16-bit compound timer 11 control status register 1 ch.1	R/W	00000000 <sub>B</sub>
0039 <sub>H</sub>	T10CR1	8/16-bit compound timer 10 control status register 1 ch.1	R/W	00000000 <sub>B</sub>
003A <sub>H</sub>	PC01	8/16-bit PPG timer 01 control register ch.0	R/W	00000000 <sub>B</sub>
003B <sub>H</sub>	PC00	8/16-bit PPG timer 00 control register ch.0	R/W	00000000 <sub>B</sub>
003C <sub>H</sub>	PC11	8/16-bit PPG timer 01 control register ch.1	R/W	00000000 <sub>B</sub>
003D <sub>H</sub>	PC10	8/16-bit PPG timer 00 control register ch.1	R/W	00000000 <sub>B</sub>
003E <sub>H</sub> to 0041 <sub>H</sub>	(Reserved Area)			
0042 <sub>H</sub>	PCNTH0	16-bit PPG status control register upper ch.0	R/W	00000000 <sub>B</sub>
0043 <sub>H</sub>	PCNTL0	16-bit PPG status control register lower ch.0	R/W	00000000 <sub>B</sub>
0044 <sub>H</sub> to 0047 <sub>H</sub>	(Reserved Area)			
0048 <sub>H</sub>	EIC00	External interrupt control register ch.0/1	R/W	00000000 <sub>B</sub>
0049 <sub>H</sub>	EIC10	External interrupt control register ch.2/3	R/W	00000000 <sub>B</sub>
004A <sub>H</sub>	EIC20	External interrupt control register ch.4/5	R/W	00000000 <sub>B</sub>
004B <sub>H</sub>	EIC30	External interrupt control register ch.6/7	R/W	00000000 <sub>B</sub>
004C <sub>H</sub> to 004F <sub>H</sub>	(Reserved Area)			
0050 <sub>H</sub>	SCR	LIN-UART serial control register	R/W	00000000 <sub>B</sub>
0051 <sub>H</sub>	SMR	LIN-UART serial mode register	R/W	00000000 <sub>B</sub>
0052 <sub>H</sub>	SSR	LIN-UART serial status register	R/W	00001000 <sub>B</sub>
0053 <sub>H</sub>	RDR/TDR	LIN-UART reception/transmission data register	R/W	00000000 <sub>B</sub>
0054 <sub>H</sub>	ESCR	LIN-UART extended status control register	R/W	00000100 <sub>B</sub>
0055 <sub>H</sub>	ECCR	LIN-UART extended communication control register	R/W	000000XX <sub>B</sub>
0056 <sub>H</sub>	SMC10	UART/SIO serial mode control register 1 ch.0	R/W	00000000 <sub>B</sub>
0057 <sub>H</sub>	SMC20	UART/SIO serial mode control register 2 ch.0	R/W	00100000 <sub>B</sub>
0058 <sub>H</sub>	SSR0	UART/SIO serial status and data register ch.0	R/W	00000001 <sub>B</sub>
0059 <sub>H</sub>	TDR0	UART/SIO serial output data register ch.0	R/W	00000000 <sub>B</sub>
005A <sub>H</sub>	RDR0	UART/SIO serial input data register ch.0	R	00000000 <sub>B</sub>

**Table A-1 MB95110B/M Series (3 / 5)**

Address	Register abbreviation	Register name	R/W	Initial value
005B <sub>H</sub> to 005F <sub>H</sub>	(Reserved Area)			
0060 <sub>H</sub>	IBCR00	I <sup>2</sup> C bus control register 0 ch.0	R/W	00000000 <sub>B</sub>
0061 <sub>H</sub>	IBCR10	I <sup>2</sup> C bus control register 1 ch.0	R/W	00000000 <sub>B</sub>
0062 <sub>H</sub>	IBSR0	I <sup>2</sup> C bus status register ch.0	R	00000000 <sub>B</sub>
0063 <sub>H</sub>	IDDR0	I <sup>2</sup> C data register ch.0	R/W	00000000 <sub>B</sub>
0064 <sub>H</sub>	IAAR0	I <sup>2</sup> C address register ch.0	R/W	00000000 <sub>B</sub>
0065 <sub>H</sub>	ICCR0	I <sup>2</sup> C clock control register ch.0	R/W	00000000 <sub>B</sub>
0066 <sub>H</sub> to 006B <sub>H</sub>	(Reserved Area)			
006C <sub>H</sub>	ADC1	8/10-bit A/D converter control register 1	R/W	00000000 <sub>B</sub>
006D <sub>H</sub>	ADC2	8/10-bit A/D converter control register 2	R/W	00000000 <sub>B</sub>
006E <sub>H</sub>	ADDH	8/10-bit A/D converter data register upper	R/W	00000000 <sub>B</sub>
006F <sub>H</sub>	ADDL	8/10-bit A/D converter data register lower	R/W	00000000 <sub>B</sub>
0070 <sub>H</sub>	WCSR	Watch counter control register	R/W	00000000 <sub>B</sub>
0071 <sub>H</sub>	(Reserved Area)			
0072 <sub>H</sub>	FSR	Flash memory status register	R/W	000x0000 <sub>B</sub>
0073 <sub>H</sub>	SWRE0	Flash memory sector write control register 0	R/W	00000000 <sub>B</sub>
0074 <sub>H</sub>	SWRE1	Flash memory sector write control register 1	R/W	00000000 <sub>B</sub>
0075 <sub>H</sub>	(Reserved Area)			
0076 <sub>H</sub>	WREN	Wild register address compare enable register	R/W	00000000 <sub>B</sub>
0077 <sub>H</sub>	WROR	Wild register data test setting register	R/W	00000000 <sub>B</sub>
0078 <sub>H</sub>	–	Register bank pointer (RP), Mirror of direct bank pointer (DP)	–	–
0079 <sub>H</sub>	ILR0	Interrupt level setting register 0	R/W	11111111 <sub>B</sub>
007A <sub>H</sub>	ILR1	Interrupt level setting register 1	R/W	11111111 <sub>B</sub>
007B <sub>H</sub>	ILR2	Interrupt level setting register 2	R/W	11111111 <sub>B</sub>
007C <sub>H</sub>	ILR3	Interrupt level setting register 3	R/W	11111111 <sub>B</sub>
007D <sub>H</sub>	ILR4	Interrupt level setting register 4	R/W	11111111 <sub>B</sub>
007E <sub>H</sub>	ILR5	Interrupt level setting register 5	R/W	11111111 <sub>B</sub>
007F <sub>H</sub>	(Reserved Area)			
0F80 <sub>H</sub>	WRARH0	Wild register address setup register upper ch.0	R/W	00000000 <sub>B</sub>
0F81 <sub>H</sub>	WRARL0	Wild register address setup register lower ch.0	R/W	00000000 <sub>B</sub>
0F82 <sub>H</sub>	WRDR0	Wild register data setup register ch.0	R/W	00000000 <sub>B</sub>
0F83 <sub>H</sub>	WRARH1	Wild register address setup register upper ch.1	R/W	00000000 <sub>B</sub>
0F84 <sub>H</sub>	WRARL1	Wild register address setup register lower ch.1	R/W	00000000 <sub>B</sub>
0F85 <sub>H</sub>	WRDR1	Wild register data setup register ch.1	R/W	00000000 <sub>B</sub>
0F86 <sub>H</sub>	WRARH2	Wild register address setup register upper ch.2	R/W	00000000 <sub>B</sub>

**Table A-1 MB95110B/M Series (4 / 5)**

Address	Register abbreviation	Register name	R/W	Initial value
0F87 <sub>H</sub>	WRARL2	Wild register address setup register lower ch.2	R/W	00000000 <sub>B</sub>
0F88 <sub>H</sub>	WRDR2	Wild register data setup register ch.2	R/W	00000000 <sub>B</sub>
0F89 <sub>H</sub> to 0F91 <sub>H</sub>	(Reserved Area)			
0F92 <sub>H</sub>	T01CR0	8/16-bit compound timer 01 control status register 0 ch.0	R/W	00000000 <sub>B</sub>
0F93 <sub>H</sub>	T00CR0	8/16-bit compound timer 00 control status register 0 ch.0	R/W	00000000 <sub>B</sub>
0F94 <sub>H</sub>	T01DR	8/16-bit compound timer 01 data register ch.0	R/W	00000000 <sub>B</sub>
0F95 <sub>H</sub>	T00DR	8/16-bit compound timer 00 data register ch.0	R/W	00000000 <sub>B</sub>
0F96 <sub>H</sub>	TMCR0	8/16-bit compound timer 00/01 timer mode control register ch.0	R/W	00000000 <sub>B</sub>
0F97 <sub>H</sub>	T11CR0	8/16-bit compound timer 11 control status register 0 ch.1	R/W	00000000 <sub>B</sub>
0F98 <sub>H</sub>	T10CR0	8/16-bit compound timer 10 control status register 0 ch.1	R/W	00000000 <sub>B</sub>
0F99 <sub>H</sub>	T11DR	8/16-bit compound timer 11 data register ch.1	R/W	00000000 <sub>B</sub>
0F9A <sub>H</sub>	T10DR	8/16-bit compound timer 10 data register ch.1	R/W	00000000 <sub>B</sub>
0F9B <sub>H</sub>	TMCR1	8/16-bit compound timer 10/11 timer mode control register ch.1	R/W	00000000 <sub>B</sub>
0F9C <sub>H</sub>	PPS01	8/16-bit PPG timer 01 cycle setting buffer register ch.0	R/W	11111111 <sub>B</sub>
0F9D <sub>H</sub>	PPS00	8/16-bit PPG timer 00 cycle setting buffer register ch.0	R/W	11111111 <sub>B</sub>
0F9E <sub>H</sub>	PDS01	8/16-bit PPG timer 01 duty setting buffer register ch.0	R/W	11111111 <sub>B</sub>
0F9F <sub>H</sub>	PDS00	8/16-bit PPG timer 00 duty setting buffer register ch.0	R/W	11111111 <sub>B</sub>
0FA0 <sub>H</sub>	PPS11	8/16-bit PPG timer 01 cycle setting buffer register ch.1	R/W	11111111 <sub>B</sub>
0FA1 <sub>H</sub>	PPS10	8/16-bit PPG timer 00 cycle setting buffer register ch.1	R/W	11111111 <sub>B</sub>
0FA2 <sub>H</sub>	PDS11	8/16-bit PPG timer 01 duty setting buffer register ch.1	R/W	11111111 <sub>B</sub>
0FA3 <sub>H</sub>	PDS10	8/16-bit PPG timer 00 duty setting buffer register ch.1	R/W	11111111 <sub>B</sub>
0FA4 <sub>H</sub>	PPGS	8/16-bit PPG startup register	R/W	00000000 <sub>B</sub>
0FA5 <sub>H</sub>	REVC	8/16-bit PPG output reverse register	R/W	00000000 <sub>B</sub>
0FA6 <sub>H</sub> to 0FA9 <sub>H</sub>	(Reserved Area)			
0FAA <sub>H</sub>	PDCRH0	16-bit PPG down counter register upper ch.0	R	00000000 <sub>B</sub>
0FAB <sub>H</sub>	PDCRL0	16-bit PPG down counter register lower ch.0	R	00000000 <sub>B</sub>
0FAC <sub>H</sub>	PCSRH0	16-bit PPG cycle setting buffer register upper ch.0	R/W	11111111 <sub>B</sub>
0FAD <sub>H</sub>	PCSRL0	16-bit PPG cycle setting buffer register lower ch.0	R/W	11111111 <sub>B</sub>
0FAE <sub>H</sub>	PDUTH0	16-bit PPG duty setting buffer register upper ch.0	R/W	11111111 <sub>B</sub>
0FAF <sub>H</sub>	PDUTL0	16-bit PPG duty setting buffer register lower ch.0	R/W	11111111 <sub>B</sub>
0FB0 <sub>H</sub> to 0FBB <sub>H</sub>	(Reserved Area)			
0FBC <sub>H</sub>	BGR1	LIN-UART baud rate generator register 1	R/W	00000000 <sub>B</sub>
0FBD <sub>H</sub>	BGR0	LIN-UART baud rate generator register 0	R/W	00000000 <sub>B</sub>
0FBE <sub>H</sub>	PSSR0	UART/SIO exclusive baud rate generator prescaler select register ch.0	R/W	00000000 <sub>B</sub>



**Table A-1 MB95110B/M Series (5 / 5)**

Address	Register abbreviation	Register name	R/W	Initial value
0FBF <sub>H</sub>	BRSR0	UART/SIO exclusive baud rate generator baud rate setting register ch.0	R/W	00000000 <sub>B</sub>
0FC0 <sub>H</sub> to 0FC2 <sub>H</sub>	(Reserved Area)			
0FC3 <sub>H</sub>	AIDRL	A/D input disable register lower	R/W	00000000 <sub>B</sub>
0FC4 <sub>H</sub> to 0FE2 <sub>H</sub>	(Reserved Area)			
0FE3 <sub>H</sub>	WCDR	Watch counter data register	R/W	00111111 <sub>B</sub>
0FE4 <sub>H</sub> to 0FE6 <sub>H</sub>	(Reserved Area)			
0FE7 <sub>H</sub>	ILSR2	Input level selection register 2	R/W	00000000 <sub>B</sub>
0FE8 <sub>H</sub> , 0FE9 <sub>H</sub>	(Reserved Area)			
0FEA <sub>H</sub>	CSVCR	Clock supervisor control register	R/W	00011100 <sub>B</sub>
0FEB <sub>H</sub> to 0FED <sub>H</sub>	(Reserved Area)			
0FEE <sub>H</sub>	ILSR	Input level selection register	R/W	00000000 <sub>B</sub>
0FEF <sub>H</sub>	WICR	Interrupt pin selection circuit control register	R/W	01000000 <sub>B</sub>
0FF0 <sub>H</sub> to 0FFF <sub>H</sub>	(Reserved Area)			

**MB95110B/M Series****APPENDIX B Table of Interrupt Causes**

This section describes the table of interrupt causes used in MB95110B/M series.

### ■ Table of Interrupt Causes

Refer to "CHAPTER 8 INTERRUPTS" for interrupt operation.

**Table B-1 MB95110B/M Series**

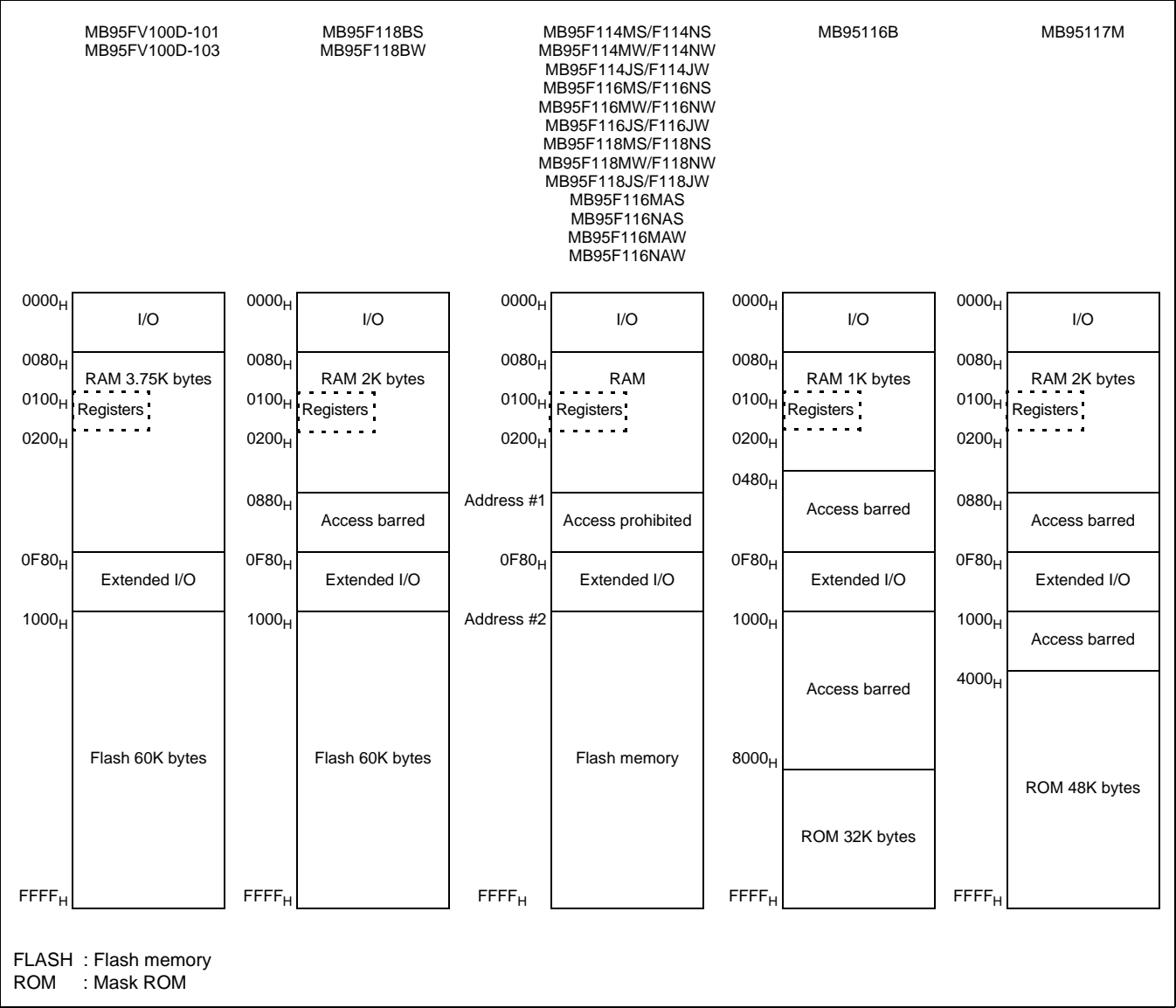
Interrupt causes	Interrupt request number	Address of vector table		Bit name of interrupt level setting register	The same level priority (Concurrence)
		Upper	Lower		
(reset vector)		FFFE <sub>H</sub>	FFFF <sub>H</sub>	-	<div>High</div> <div>↑</div> <div>↓</div> <div>Low</div>
(mode data)		-	FFFD <sub>H</sub>	-	
External interrupt ch.0	IRQ0	FFFA <sub>H</sub>	FFFB <sub>H</sub>	L00 [1:0]	
External interrupt ch.4					
External interrupt ch.1	IRQ1	FFF8 <sub>H</sub>	FFF9 <sub>H</sub>	L01 [1:0]	
External interrupt ch.5					
External interrupt ch.2	IRQ2	FFF6 <sub>H</sub>	FFF7 <sub>H</sub>	L02 [1:0]	
External interrupt ch.6					
External interrupt ch.3	IRQ3	FFF4 <sub>H</sub>	FFF5 <sub>H</sub>	L03 [1:0]	
External interrupt ch.7					
UART/SIO ch.0	IRQ4	FFF2 <sub>H</sub>	FFF3 <sub>H</sub>	L04 [1:0]	
8/16-bit compound timer ch.0 (lower)	IRQ5	FFF0 <sub>H</sub>	FFF1 <sub>H</sub>	L05 [1:0]	
8/16-bit compound timer ch.0 (upper)	IRQ6	FFEE <sub>H</sub>	FFEF <sub>H</sub>	L06 [1:0]	
LIN-UART (reception)	IRQ7	FFEC <sub>H</sub>	FFED <sub>H</sub>	L07 [1:0]	
LIN-UART (transmission)	IRQ8	FFEA <sub>H</sub>	FFEB <sub>H</sub>	L08 [1:0]	
8/16-bit PPG ch.1 (lower)	IRQ9	FFE8 <sub>H</sub>	FFE9 <sub>H</sub>	L09 [1:0]	
8/16-bit PPG ch.1 (upper)	IRQ10	FFE6 <sub>H</sub>	FFE7 <sub>H</sub>	L10 [1:0]	
(Not used)	IRQ11	FFE4 <sub>H</sub>	FFE5 <sub>H</sub>	L11 [1:0]	
8/16-bit PPG ch.0 (lower)	IRQ12	FFE2 <sub>H</sub>	FFE3 <sub>H</sub>	L12 [1:0]	
8/16-bit PPG ch.0 (upper)	IRQ13	FFE0 <sub>H</sub>	FFE1 <sub>H</sub>	L13 [1:0]	
8/16-bit compound timer ch.1 (upper)	IRQ14	FFDE <sub>H</sub>	FFDF <sub>H</sub>	L14 [1:0]	
16-bit PPG ch.0	IRQ15	FFDC <sub>H</sub>	FFDD <sub>H</sub>	L15 [1:0]	
I <sup>2</sup> C ch.0	IRQ16	FFDA <sub>H</sub>	FFDB <sub>H</sub>	L16 [1:0]	
(Not used)	IRQ17	FFD8 <sub>H</sub>	FFD9 <sub>H</sub>	L17 [1:0]	
8/10-bit A/D	IRQ18	FFD6 <sub>H</sub>	FFD7 <sub>H</sub>	L18 [1:0]	
Time-base timer	IRQ19	FFD4 <sub>H</sub>	FFD5 <sub>H</sub>	L19 [1:0]	
Watch prescaler/counter	IRQ20	FFD2 <sub>H</sub>	FFD3 <sub>H</sub>	L20 [1:0]	
(Not used)	IRQ21	FFD0 <sub>H</sub>	FFD1 <sub>H</sub>	L21 [1:0]	
8/16-bit compound timer ch.1 (lower)	IRQ22	FFCE <sub>H</sub>	FFCF <sub>H</sub>	L22 [1:0]	
Flash memory	IRQ23	FFCC <sub>H</sub>	FFCD <sub>H</sub>	L23 [1:0]	

APPENDIX C Memory Map

This section shows the memory map of MB95110B/M series.

Memory Map

Figure C-1 Memory Map



# MB95110B/M Series

	Flash memory	RAM	Address #1	Address #2
MB95F114MS/F114NS	16K bytes	512 bytes	0280 <sub>H</sub>	C000 <sub>H</sub>
MB95F114MW/F114NW				
MB95F114JS/F114JW				
MB95F116MS/F116NS	32K bytes	1K byte	0480 <sub>H</sub>	8000 <sub>H</sub>
MB95F116MW/F116NW				
MB95F116JS/F116JW				
MB95F116MAS/F116NAS				
MB95F116MAW/F116NAW				
MB95F118MS/F118NS	60K bytes	2K bytes	0880 <sub>H</sub>	1000 <sub>H</sub>
MB95F118MW/F118NW				
MB95F118JS/F118JW				

## APPENDIX D Pin Status of MB95110B/M series

The state of the pin of the MB95110B/M series in each mode is shown in Table D-1.

### ■ Pin Status in Each Mode

**Table D-1 Pin Status in Each Mode (1 / 2)**

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		While resetting
			SPL=0	SPL=1	SPL=0	SPL=1	
X0	Oscillation circuit input	Oscillation circuit input	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Oscillation circuit input
X1	Oscillation circuit output	Oscillation circuit input	"H"	"H"	"H"	"H"	Oscillation circuit output
MOD	Mode input	Mode input	Mode input	Mode input	Mode input	Mode input	Mode input
RST	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input	Reset input
P00/INT00	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z input interception (However, an external interrupt can be input when the external interrupt is enable. )	I/O port/ peripheral function I/O	Hi-Z input interception (However, an external interrupt can be input when the external interrupt is enable. )	Hi-Z Input enable* <sup>1</sup> (However, it doesn't function. )
P01/INT01							
P02/INT02							
P03/INT03							
P04/INT04							
P05/INT05							
P06/INT06							
P07/INT07							
P10/UI0	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	Hi-Z Input enable* <sup>1</sup> (However, it doesn't function. )
P11/UO0							
P12/UCK0							
P13/TRG0/ ADTG							
P14/PPG0							
P15							
P20/PPG00	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	Hi-Z Input enable* <sup>1</sup> (However, it doesn't function. )
P21/PPG01							
P22/TO00							
P23/TO01							
P24/EC0							

**MB95110B/M Series****Table D-1 Pin Status in Each Mode (2 / 2)**

Pin name	Normal operation	Sleep mode	Stop mode		Watch mode		While resetting
			SPL=0	SPL=1	SPL=0	SPL=1	
P30/AN00	I/O port/ Analog input	I/O port/ Analog input	I/O port/ Analog input	Hi-Z (However, the setting of the pull-up is effective.) Input interception	I/O port/ Analog input	Hi-Z (However, the setting of the pull-up is effective.) Input interception	Hi-Z Input disable <sup>*2</sup>
P31/AN01							
P32/AN02							
P33/AN03							
P34/AN04							
P35/AN05							
P36/AN06							
P37/AN07							
P50/SCL0	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	I/O port/ peripheral function I/O	Hi-Z (However, the setting of the pull-up is effective.) Input interception	Hi-Z Input enable (However, it doesn't function. )
P51/SDA0							
P60/PPG10	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	I/O port/ peripheral function I/O	Hi-Z Input interception	I/O port/ peripheral function I/O	Hi-Z Input interception	Hi-Z Input enable <sup>*1</sup> (However, it doesn't function. )
P61/PPG11							
P62/TO10							
P63/TO11							
P64/EC1							
P65/SCK							
P66/SOT							
P67/SIN							
PG0/C <sup>*4</sup>	I/O port	I/O port	I/O port	Hi-Z Input interception	I/O port	Hi-Z Input interception	Hi-Z Input enable <sup>*1</sup> (However, it doesn't function. )
PG1/X0A <sup>*3</sup>							
PG2/X1A <sup>*3</sup>							

SPL: Pin status specification bit of standby control register (STBC: SPL)

Hi-Z: High impedance

\*1: "Input enable" means the input function is possible. Therefore, it is necessary to process the pull-up and the pull-down or to prevent leakage being generated by the input from the outside. It is the same status as other ports when using it as an output port.

\*2: "Input disable" means direct input gate operation from the pin is disable status.

\*3: These pins status becomes the sub clock input and the sub clock output in dual clock products.

\*4: For the 5V product, the C pin is used.

## APPENDIX E Instruction Overview

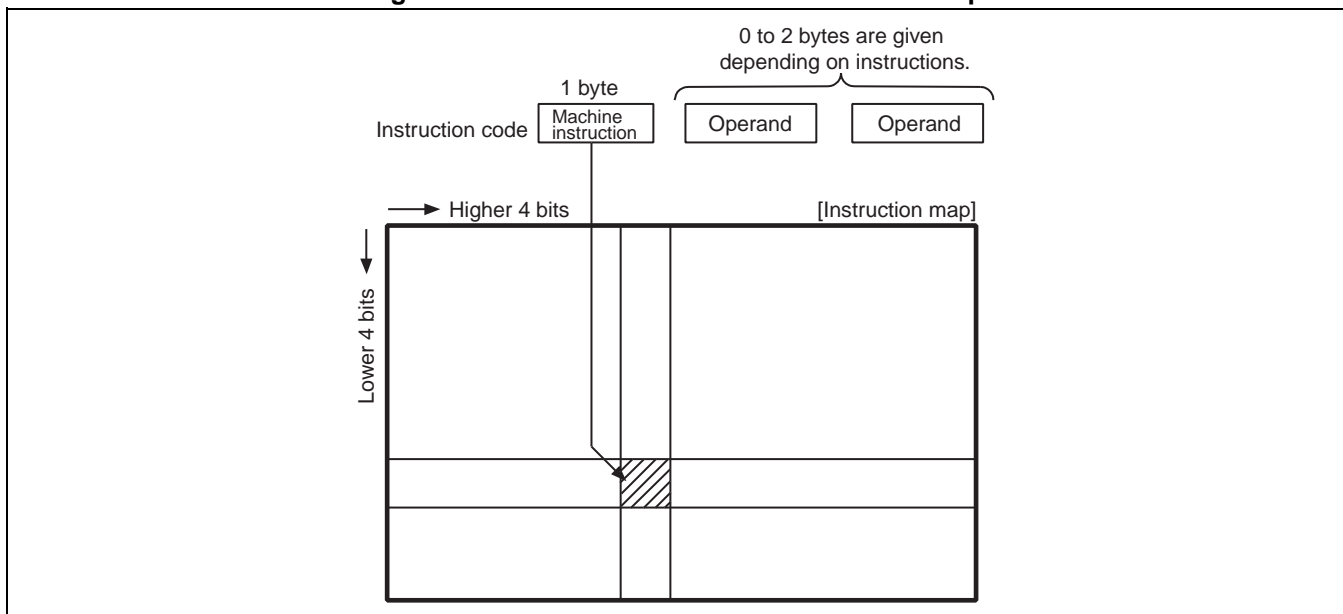
This section explains the instructions used in F<sup>2</sup>MC-8FX.

### ■ Instruction Overview of F<sup>2</sup>MC-8FX

In F<sup>2</sup>MC-8FX, there are 140 kinds of one byte machine instructions (as the map, 256 bytes), and the instruction code is composed of the instruction and the operand following it.

Figure E-1 shows the correspondence of the instruction code and the instruction map.

Figure E-1 Instruction Code and Instruction Map



- The instruction is classified into following four types; forwarding system, operation system, branch system and others.
- There are various methods of addressing, and ten kinds of addressing can be selected by the selection and the operand specification of the instruction.
- This provides with the bit operation instruction, and can operate the read modification write.
- There is an instruction that directs special operation.

# MB95110B/M Series

## ■ Explanation of Display Sign of Instruction

Table E-1 shows the explanation of the sign used by explaining the instruction code of this APPENDIX E.

**Table E-1 Explanation of Sign in Instruction Table**

Sign	Signification
dir	Direct address (8-bit length)
off	Offset (8-bit length)
ext	Extended address (16-bit length)
#vct	Vector table number (3-bit length)
#d8	Immediate data (8-bit length)
#d16	Immediate data (16-bit length)
dir:b	Bit direct address (8-bit length: 3-bit length)
rel	Branch relative address (8-bit length)
@	Register indirect (Example: @A, @IX, @EP)
A	Accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
AH	Upper 8-bit of accumulator (8-bit length)
AL	Lower 8-bit of accumulator (8-bit length)
T	Temporary accumulator (Whether 8- bit length or 16- bit length is decided by the instruction used.)
TH	Upper 8-bit of temporary accumulator (8-bit length)
TL	Lower 8-bit of temporary accumulator (8-bit length)
IX	Index register (16-bit length)
EP	Extra pointer (16-bit length)
PC	Program counter (16-bit length)
SP	Stack pointer (16-bit length)
PS	Program status (16-bit length)
dr	Either of accumulator or index register (16-bit length)
CCR	Condition code register (8-bit length)
RP	Register bank pointer (5-bit length)
DP	Direct bank pointer (3-bit length)
Ri	General-purpose register (8-bit length, i = 0 to 7)
x	This shows that x is immediate data. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
(x)	This shows that contents of x are objects of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)
((x))	This shows that the address that contents of x show is an object of the access. (Whether 8- bit length or 16- bit length is decided by the instruction used.)



■ **Explanation of Item in Instruction Table**

**Table E-2 Explanation of Item in Instruction Table**

Item	Description
MNEMONIC	It shows the assembly description of the instruction.
~	It shows the number of cycles of the instruction. One instruction cycle is a machine cycle. Note: The number of cycles of the instruction can be delayed by 1 cycle by the immediately preceding instruction. Moreover, the number of cycles of the instruction might be extended in the access to the I/O area.
#	It shows the number of bytes for the instruction.
Operation	It shows the operations for the instruction.
TL, TH, AH	They show the change (auto forwarding from A to T) in the content when each TL, TH, and AH instruction is executed. The sign in the column indicates the followings respectively. <ul style="list-style-type: none"> <li>• -: No change</li> <li>• dH: upper 8 bits of the data described in operation.</li> <li>• AL and AH: the contents become those of the immediately preceding instruction's AL and AH.</li> <li>• 00: Become 00</li> </ul>
N, Z, V, C	They show the instruction into which the corresponding flag is changed respectively. The sign in the column shows the followings respectively. <ul style="list-style-type: none"> <li>• -: No change</li> <li>• +: Change</li> <li>• R: Become "0"</li> <li>• S: Become "1"</li> </ul>
OP CODE	It shows the code of the instruction. When a pertinent instruction occupies two or more codes, it follows the following description rules. [Example] 48 to 4F: This shows 48, 49....4F.

# MB95110B/M Series

## E.1 Addressing

F<sup>2</sup>MC-8FX has the following ten types of addressings:

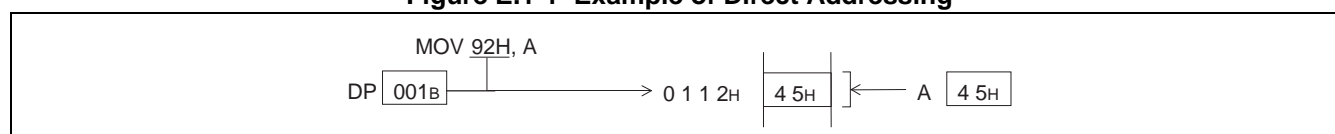
- Direct addressing
- Extended addressing
- Bit direct addressing
- Index addressing
- Pointer addressing
- General-purpose register addressing
- Immediate addressing
- Vector addressing
- Relative addressing
- Inherent addressing

### ■ Explanation of Addressing

#### ● Direct addressing

This is used when accessing the direct area of "0000<sub>H</sub>" to "047F<sub>H</sub>" with addressing indicated "dir" in instruction table. In this addressing, when the operand address is "00<sub>H</sub>" to "7F<sub>H</sub>", it is accessed into "0000<sub>H</sub>" to "007F<sub>H</sub>". Moreover, when the operand address is "80<sub>H</sub>" to "FF<sub>H</sub>", the access can be mapped in "0080<sub>H</sub>" to "047F<sub>H</sub>" by setting of direct bank pointer DP. Figure E.1-1 shows an example.

**Figure E.1-1 Example of Direct Addressing**

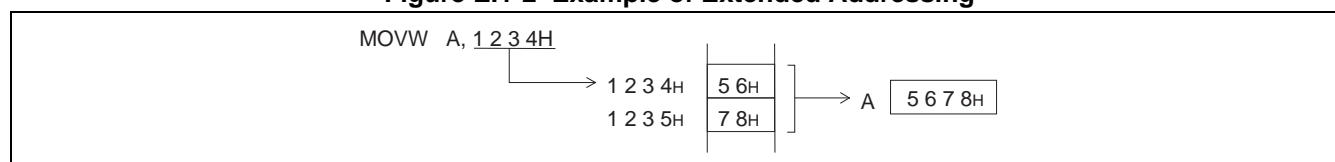


#### ● Extended addressing

This is used when the area of the entire 64 K bytes is accessed by addressing shown "ext" in the instruction table. In this addressing, the first operand specifies one high rank byte of the address and the second operand specifies one subordinate position byte of the address.

Figure E.1-2 shows an example.

**Figure E.1-2 Example of Extended Addressing**

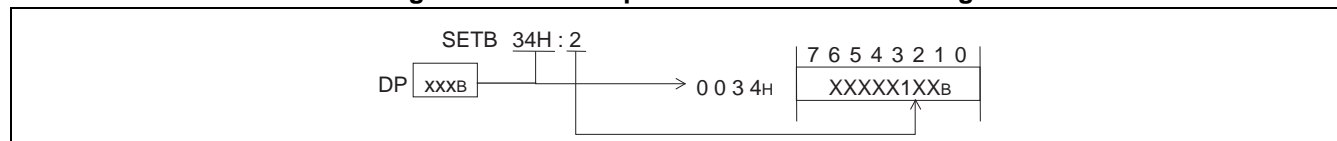


● Bit direct addressing

This is used when accessing the direct area of "0000<sub>H</sub>" to "047F<sub>H</sub>" in bit unit with addressing indicated "dir:b" in instruction table. In this addressing, when the operand address is "00<sub>H</sub>" to "7F<sub>H</sub>", it is accessed into "0000<sub>H</sub>" to "007F<sub>H</sub>". Moreover, when the operand address is "80<sub>H</sub>" to "FF<sub>H</sub>", the access can be mapped in "0080<sub>H</sub>" to "047F<sub>H</sub>" by setting of direct bank pointer DP. The position of the bit in the specified address is specified by the values of the instruction code of three subordinate position bits.

Figure E.1-3 shows an example.

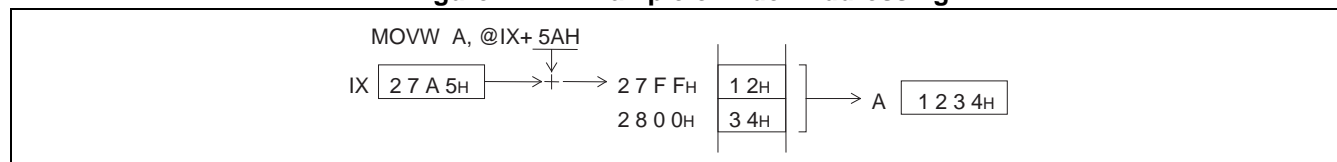
**Figure E.1-3 Example of Bit Direct Addressing**



● Index addressing

This is used when the area of the entire 64 K bytes is accessed by addressing shown "@IX+off" in the instruction table. In this addressing, the content of the first operand is sign extended and added to IX (index register) to the resulting address. Figure E.1-4 shows an example.

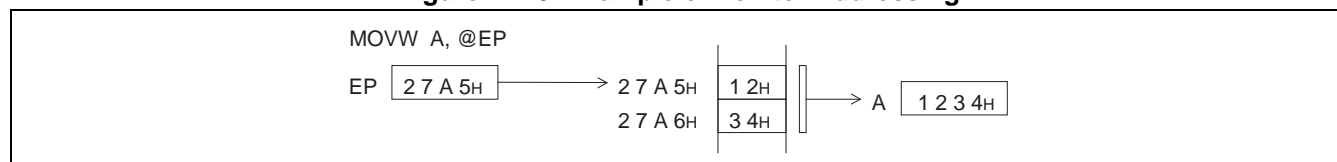
**Figure E.1-4 Example of Index Addressing**



● Pointer addressing

This is used when the area of the entire 64 K bytes is accessed by addressing shown "@EP" in the instruction table. In this addressing, the content of EP (extra pointer) is assumed to be an address. Figure E.1-5 shows an example.

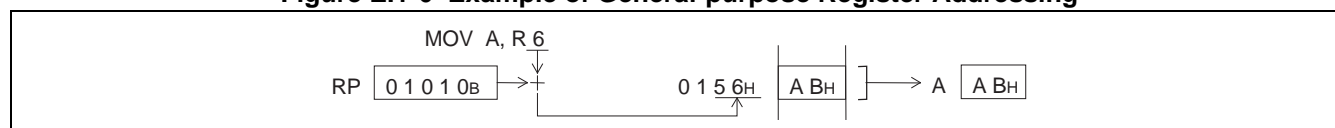
**Figure E.1-5 Example of Pointer Addressing**



● General-purpose register addressing

This is used when accessing the register bank in general-purpose register area with the addressing shown "Ri" in instruction table. In this addressing, fix one high rank byte of the address to "01" and create one subordinate position byte from the contents of RP (register bank pointer) and three subordinate bits of the operation code to access to this address. Figure E.1-6 shows an example.

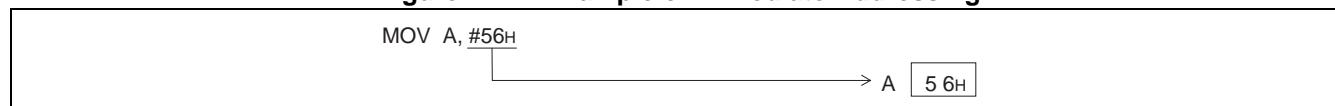
**Figure E.1-6 Example of General-purpose Register Addressing**



● Immediate addressing

This is used when immediate data is needed in addressing shown "#d8" in the instruction table. In this addressing, the operand becomes immediate data as it is. The specification of byte/word depends on the operation code. Figure E.1-7 shows an example.

**Figure E.1-7 Example of Immediate Addressing**



● Vector addressing

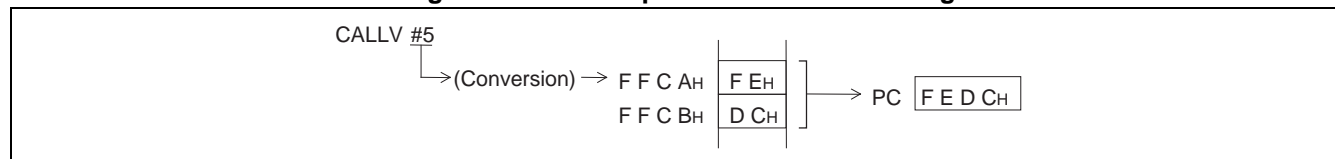
This is used when branching to the subroutine address registered in the table with the addressing shown "#vct" in the instruction table. In this addressing, information on "#vct" is contained in the operation code, and the address of the table is created using the combinations shown in Table E.1-1.

**Table E.1-1 Vector Table Address Corresponding to "#vct"**

#vct	Vector table address (jump destination high-ranking address: subordinate address)
0	FFC0 <sub>H</sub> : FFC1 <sub>H</sub>
1	FFC2 <sub>H</sub> : FFC3 <sub>H</sub>
2	FFC4 <sub>H</sub> : FFC5 <sub>H</sub>
3	FFC6 <sub>H</sub> : FFC7 <sub>H</sub>
4	FFC8 <sub>H</sub> : FFC9 <sub>H</sub>
5	FFCA <sub>H</sub> : FFCE <sub>H</sub>
6	FFCC <sub>H</sub> : FFCD <sub>H</sub>
7	FFCE <sub>H</sub> : FFCF <sub>H</sub>

Figure E.1-8 shows an example.

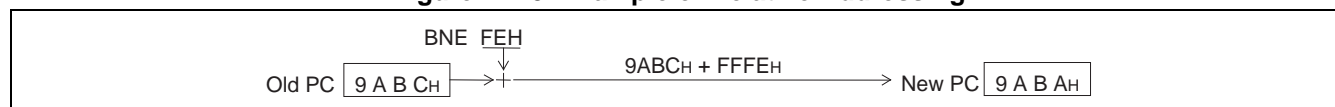
**Figure E.1-8 Example of Vector Addressing**



● **Relative addressing**

This is used when branching to the area in 128 bytes before and behind PC (program counter) with the addressing shown "rel" in the instruction table. In this addressing, add the content of the operand to PC with the sign and store the result in PC. Figure E.1-9 shows an example.

**Figure E.1-9 Example of Relative Addressing**

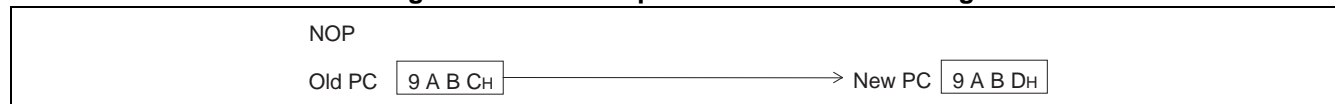


In this example, by jumping to the address where the operation code of BNE is stored, it results in an infinite loop.

● **Inherent addressing**

This is used when doing the operation decided by the operation code with the addressing that does not have the operand in the instruction table. In this addressing, the operation depends on each instruction. Figure E.1-10 shows an example.

**Figure E.1-10 Example of Inherent Addressing**



# MB95110B/M Series

## E.2 Special Instruction

This section explains special instructions other than the addressings.

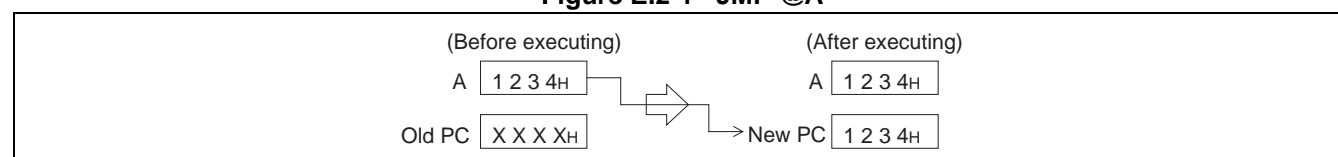
### ■ Special Instruction

#### ● JMP @A

This instruction is to branch the content of A (accumulator) to PC (program counter) as an address. N pieces of the jump destination is arranged on the table, and one of the contents is selected and transferred to A. N branch processing can be done by executing this instruction.

Figure E.2-1 shows a summary of the instruction.

**Figure E.2-1 JMP @A**

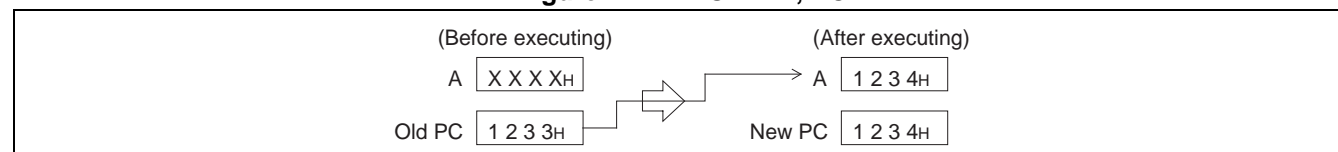


#### ● MOVW A, PC

This instruction works as the opposite of "JMP @A". That is, it stores the content of PC to A. When you have executed this instruction in the main routine and set it to call a specific subroutine, you can make sure that the content of A is the specified value in the subroutine. Also, you can identify that the branch is not from the part that cannot be expected, and use it for the reckless driving judgment.

Figure E.2-2 shows a summary of the instruction.

**Figure E.2-2 MOVW A, PC**



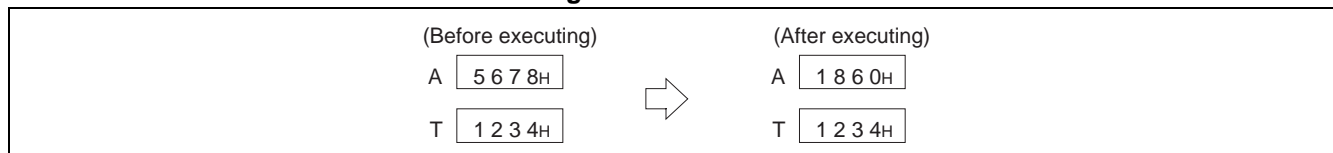
When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-2, the value "1234<sub>H</sub>" stored in A corresponds to the address where the following operation code of "MOVW A, PC" is stored.

#### ● MULU A

This instruction performs an unsigned multiplication of AL (lower 8-bit of the accumulator) and TL (lower 8-bit of the temporary accumulator), and stores the 16-bit result in A. The contents of T (temporary accumulator) do not change. The contents of AH (higher 8-bit of the accumulator) and TH (higher 8-bit of the temporary accumulator) before execution of the instruction are not used for the operation. The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a multiplication.

Figure E.2-3 shows a summary of the instruction.

**Figure E.2-3 MULU A**



● **DIVU A**

This instruction divides the 16-bit value in T by the unsigned 16-bit value in A, and stores the 16-bit result and the 16-bit remainder in A and T, respectively. When the value in A before execution of instruction is "0", the Z flag becomes "1" to indicate zero-division is executed. The instruction does not change other flags, and therefore care must be taken when a branch may occur depending on the result of a division.

Figure E.2-4 shows a summary of the instruction.

**Figure E.2-4 DIVU A**

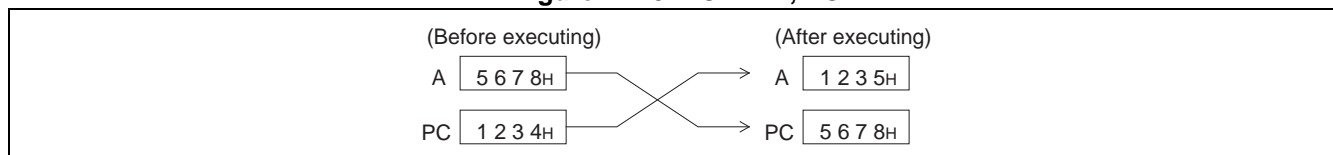


● **XCHW A, PC**

This instruction swaps the contents of A and PC, resulting in a branch to the address contained in A before execution of the instruction. After the instruction is executed, A becomes the address that follows the address where the operation code of "XCHW A, PC" is stored. This instruction is effective especially when it is used in the main routine to specify a table for use in a subroutine.

Figure E.2-5 shows a summary of the instruction.

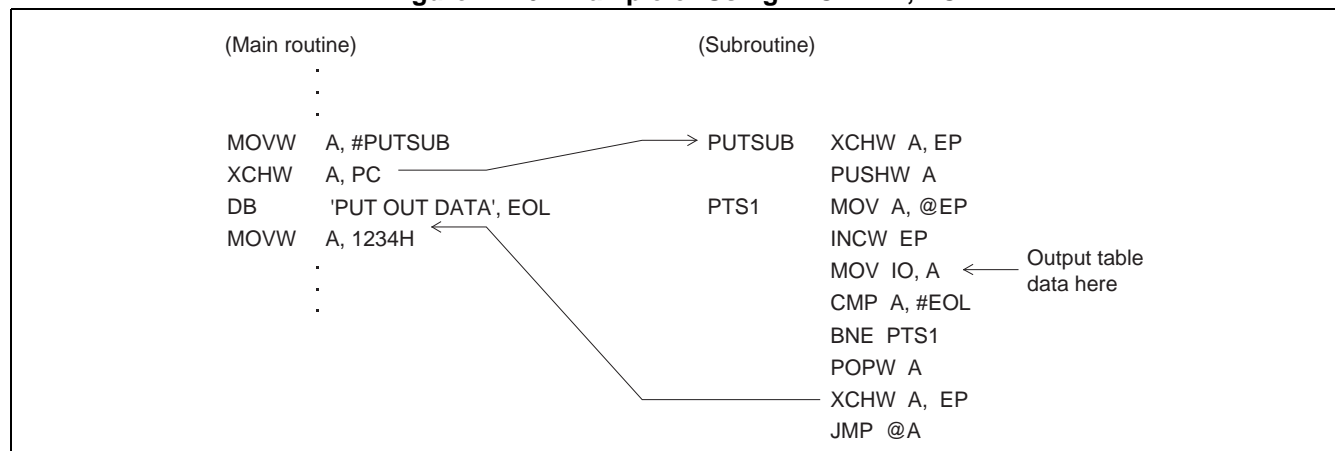
**Figure E.2-5 XCHW A, PC**



When this instruction is executed, the content of A reaches the same value as the address where the following instruction is stored, rather than the address where operation code of this instruction is stored. Therefore, in Figure E.2-5, the value "1235<sub>H</sub>" stored in A corresponds to the address where the following operation code of "XCHW A, PC" is stored. This is why "1235<sub>H</sub>" is stored instead of "1234<sub>H</sub>".

Figure E.2-6 shows an assembler language example.

Figure E.2-6 Example of Using "XCHW A, PC"

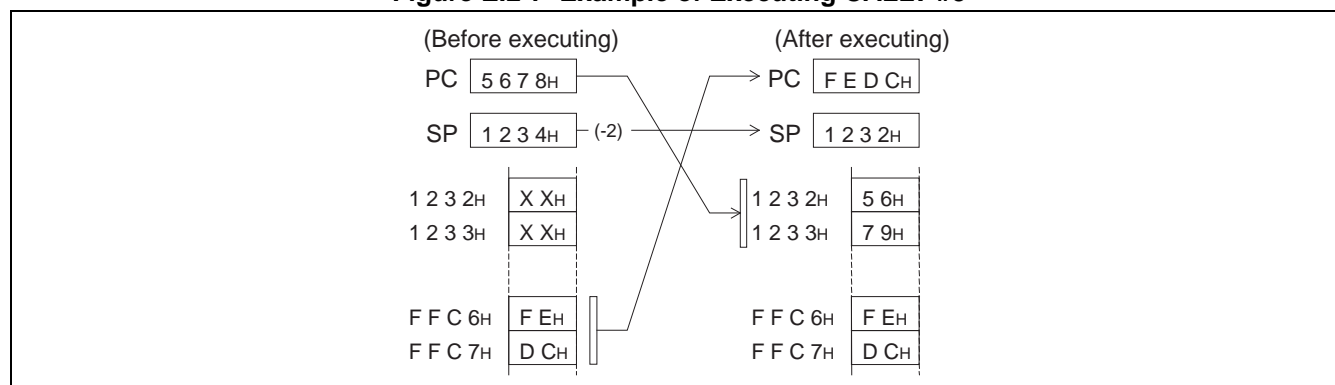


### ● CALLV #vct

This instruction is used to branch to a subroutine address stored in the vector table. The instruction saves the return address (contents of PC) in the location at the address contained in SP (stack pointer), and uses vector addressing to cause a branch to the address stored in the vector table. Because `CALLV #vct` is a 1-byte instruction, the use of this instruction for frequently used subroutines can reduce the entire program size.

Figure E.2-7 shows a summary of the instruction.

Figure E.2-7 Example of Executing CALLV #3



After the `CALLV #vct` instruction is executed, the contents of PC saved on the stack area are the address of the operation code of the next instruction, rather than the address of the operation code of `CALLV #vct`. Accordingly, Figure E.2-7 shows that the value saved in the stack (1232<sub>H</sub> and 1233<sub>H</sub>) is 5679<sub>H</sub>, which is the address of the operation code of the instruction that follows "CALLV #vct" (return address).



**Table E.2-1 Vector Table**

Vector use (call instruction)	Vector table address	
	Upper	Lower
CALLV #7	FFCE <sub>H</sub>	FFCF <sub>H</sub>
CALLV #6	FFCC <sub>H</sub>	FFCD <sub>H</sub>
CALLV #5	FFCA <sub>H</sub>	FFCB <sub>H</sub>
CALLV #4	FFC8 <sub>H</sub>	FFC9 <sub>H</sub>
CALLV #3	FFC6 <sub>H</sub>	FFC7 <sub>H</sub>
CALLV #2	FFC4 <sub>H</sub>	FFC5 <sub>H</sub>
CALLV #1	FFC2 <sub>H</sub>	FFC3 <sub>H</sub>
CALLV #0	FFC0 <sub>H</sub>	FFC1 <sub>H</sub>

# MB95110B/M Series

## E.3 Bit Manipulation Instructions (SETB, CLRB)

Some peripheral function registers include bits that are read differently than usual by a bit manipulation instruction.

### ■ Read-modify-write Operation

By using these bit manipulation instructions, you can set only the specified bit in a register or RAM location to "1" (SETB) or clear to "0" (CLRB). However, as the CPU operates data in 8-bit units, the actual operation (read-modify-write operation) involves a sequence of steps: 8-bit data is read, the specified bit is changed, and the data is written back to the location at the original address.

Table E.3-1 shows bus operation for bit manipulation instructions.

**Table E.3-1 Bus Operation for Bit Manipulation Instructions**

CODE	MNEMONIC	~	Cycle	Address bus	Data bus	RD	WR	RMW
A0 to A7	CLRB dir:b	4	1	N+2	Next instruction	1	0	1
			2	dir address	Data	1	0	1
A8 to AF	SETB dir:b		3	dir address	Data	0	1	0
			4	N+3	Instruction after next	1	0	0

### ■ Read Destination on the Execution of Bit Manipulation Instructions

For some I/O ports and the interrupt request flag bits, the read destination differs between a normal read operation and a read-modify-write operation.

#### ● I/O ports (during a bit manipulation)

From some I/O ports, an I/O pin value is read during a normal read operation, while a port data register value is read during a bit manipulation. This prevents the other port data register bits from being changed accidentally, regardless of the I/O directions and states of the pins.

#### ● Interrupt request flag bits (during a bit manipulation)

An interrupt request flag bit functions as a flag bit indicating whether an interrupt request exists during a normal read operation, however, "1" is always read from this bit during a bit manipulation. This prevents the flag from being cleared accidentally by writing the value "0" to the interrupt request flag bit when manipulating another bit.

## E.4 F<sup>2</sup>MC-8FX Instructions

Table E.4-1 to Table E.4-4 show the instructions used by the F<sup>2</sup>MC-8FX.

### ■ Transfer Instructions

Table E.4-1 Transfer Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	MOV dir, A	3	2	(dir) ← (A)	-	-	-	-	-	-	-	45
2	MOV @IX + off, A	3	2	((IX) + off) ← (A)	-	-	-	-	-	-	-	46
3	MOV ext, A	4	3	(ext) ← (A)	-	-	-	-	-	-	-	61
4	MOV @EP, A	2	1	((EP)) ← (A)	-	-	-	-	-	-	-	47
5	MOV Ri, A	2	1	(Ri) ← (A)	-	-	-	-	-	-	-	48 to 4F
6	MOV A, #d8	2	2	(A) ← d8	AL	-	-	+	+	-	-	04
7	MOV A, dir	3	2	(A) ← (dir)	AL	-	-	+	+	-	-	05
8	MOV A, @IX + off	3	2	(A) ← ((IX) - off)	AL	-	-	+	+	-	-	06
9	MOV A, ext	4	3	(A) ← (ext)	AL	-	-	+	+	-	-	60
10	MOV A, @A	2	1	(A) ← ((A))	AL	-	-	+	+	-	-	92
11	MOV A, @EP	2	1	(A) ← ((EP))	AL	-	-	+	+	-	-	07
12	MOV A, Ri	2	1	(A) ← (Ri)	AL	-	-	-	+	-	-	08 to 0F
13	MOV dir, #d8	4	3	(dir) ← d8	-	-	-	-	-	-	-	85
14	MOV @IX + off, #d8	4	3	((IX) + off) ← d8	-	-	-	-	-	-	-	86
15	MOV @EP, #d8	3	2	((EP)) ← d8	-	-	-	-	-	-	-	87
16	MOV Ri, #d8	3	2	(Ri) ← d8	-	-	-	-	-	-	-	88 to 8F
17	MOVW dir, A	4	2	(dir) ← (AH), (dir + 1) ← (AL)	-	-	-	-	-	-	-	D5
18	MOVW @IX + off, A	4	2	((IX) + off) ← (AH), ((IX) + off + 1) ← (AL)	-	-	-	-	-	-	-	D6
19	MOVW ext, A	5	3	(ext) ← (AH), (ext + 1) ← (AL)	-	-	-	-	-	-	-	D4
20	MOVW @EP, A	3	1	((EP)) ← (AH), ((EP) + 1) ← (AL)	-	-	-	-	-	-	-	D7
21	MOVW EP, A	1	1	(EP) ← (A)	-	-	-	-	-	-	-	E3
22	MOVW A, #d16	3	3	(A) ← d16	AL	AH	dH	+	+	-	-	E4
23	MOVW A, dir	4	2	(AH) ← (dir), (AL) ← (dir + 1)	AL	AH	dH	+	+	-	-	C5
24	MOVW A, @IX + off	4	2	(AH) ← ((IX) + off), (AL) ← ((IX) + off + 1)	AL	AH	dH	-	+	-	-	C6
25	MOVW A, ext	5	3	(AH) ← (ext), (AL) ← (ext + 1)	AL	AH	dH	+	+	-	-	C4
26	MOVW A, @A	3	1	(AH) ← ((A)), (AL) ← ((A) + 1)	AL	AH	dH	+	+	-	-	93
27	MOVW A, @EP	3	1	(AH) ← ((EP)), (AL) ← ((EP) + 1)	AL	AH	dH	-	+	-	-	C7
28	MOVW A, EP	1	1	(A) ← (EP)	-	-	dH	-	-	-	-	F3
29	MOVW EP, #d16	3	3	(EP) ← d16	-	-	-	-	-	-	-	E7
30	MOVW IX, A	1	1	(IX) ← (A)	-	-	-	-	-	-	-	E2
31	MOVW A, IX	1	1	(A) ← (IX)	-	-	dH	-	-	-	-	F2
32	MOVW SP, A	1	1	(SP) ← (A)	-	-	-	-	-	-	-	E1
33	MOVW A, SP	1	1	(A) ← (SP)	-	-	dH	-	-	-	-	F1
34	MOV @A, T	2	1	((A)) ← (T)	-	-	-	-	-	-	-	82
35	MOVW @A, T	3	1	((A)) ← (TH), ((A) + 1) ← (TL)	-	-	-	-	-	-	-	83
36	MOVW IX, #d16	3	3	(IX) ← d16	-	-	-	-	-	-	-	E6
37	MOVW A, PS	1	1	(A) ← (PS)	-	-	dH	-	-	-	-	70
38	MOVW PS, A	1	1	(PS) ← (A)	-	-	-	+	+	-	+	71
39	MOVW SP, #d16	3	3	(SP) ← d16	-	-	-	-	-	-	-	E5
40	SWAP	1	1	(AH) ↔ (AL)	-	-	AL	-	-	-	-	10
41	SETB dir:b	4	2	(dir) : b ← 1	-	-	-	-	-	-	-	A8 to AF
42	CLRB dir:b	4	2	(dir) : b ← 0	-	-	-	-	-	-	-	A0 to A7
43	XCH A, T	1	1	(AL) ↔ (TL)	AL	-	-	-	-	-	-	42
44	XCHW A, T	1	1	(A) ↔ (T)	AL	AH	dH	-	-	-	-	43
45	XCHW A, EP	1	1	(A) ↔ (EP)	-	-	dH	-	-	-	-	F7
46	XCHW A, IX	1	1	(A) ↔ (IX)	-	-	dH	-	-	-	-	F6
47	XCHW A, SP	1	1	(A) ↔ (SP)	-	-	dH	-	-	-	-	F5
48	MOVW A, PC	2	1	(A) ← (PC)	-	-	dH	-	-	-	-	F0

## Note:

In automatic transfer to T during byte transfer to A, AL is transferred to TL.  
If an instruction has plural operands, they are saved in the order indicated by MNEMONIC.

## ■ Arithmetic Operation Instructions

Table E.4-2 Arithmetic Operation Instruction (1 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	ADDC A, Ri	2	1	$(A) \leftarrow (A) + (Ri) + C$	-	-	-	+	+	+	+	28 to 2F
2	ADDC A, #d8	2	2	$(A) \leftarrow (A) + d8 + C$	-	-	-	+	+	+	+	24
3	ADDC A, dir	3	2	$(A) \leftarrow (A) + (dir) + C$	-	-	-	+	+	+	+	25
4	ADDC A, @IX + off	3	2	$(A) \leftarrow (A) + ((IX) + off) + C$	-	-	-	+	+	+	+	26
5	ADDC A, @EP	2	1	$(A) \leftarrow (A) + ((EP)) + C$	-	-	-	+	+	+	+	27
6	ADDCW A	1	1	$(A) \leftarrow (A) + (T) + C$	-	-	dH	+	+	+	+	23
7	ADDC A	1	1	$(AL) \leftarrow (AL) + (TL) + C$	-	-	-	+	+	+	+	22
8	SUBC A, Ri	2	1	$(A) \leftarrow (A) - (Ri) - C$	-	-	-	+	+	+	+	38 to 3F
9	SUBC A, #d8	2	2	$(A) \leftarrow (A) - d8 - C$	-	-	-	+	+	+	+	34
10	SUBC A, dir	3	2	$(A) \leftarrow (A) - (dir) - C$	-	-	-	+	+	+	+	35
11	SUBC A, @IX + off	3	2	$(A) \leftarrow (A) - ((IX) + off) - C$	-	-	-	+	+	+	+	36
12	SUBC A, @EP	2	1	$(A) \leftarrow (A) - ((EP)) - C$	-	-	-	+	+	+	+	37
13	SUBCW A	1	1	$(A) \leftarrow (T) - (A) - C$	-	-	dH	+	+	+	+	33
14	SUBC A	1	1	$(AL) \leftarrow (TL) - (AL) - C$	-	-	-	+	+	+	+	32
15	INC Ri	3	1	$(Ri) \leftarrow (Ri) + 1$	-	-	-	+	+	+	-	C8 to CF
16	INCW EP	1	1	$(EP) \leftarrow (EP) + 1$	-	-	-	-	-	-	-	C3
17	INCW IX	1	1	$(IX) \leftarrow (IX) + 1$	-	-	-	-	-	-	-	C2
18	INCW A	1	1	$(A) \leftarrow (A) + 1$	-	-	dH	+	+	-	-	C0
19	DEC Ri	3	1	$(Ri) \leftarrow (Ri) - 1$	-	-	-	+	+	+	-	D8 to DF
20	DECW EP	1	1	$(EP) \leftarrow (EP) - 1$	-	-	-	-	-	-	-	D3
21	DECW IX	1	1	$(IX) \leftarrow (IX) - 1$	-	-	-	-	-	-	-	D2
22	DECW A	1	1	$(A) \leftarrow (A) - 1$	-	-	dH	+	+	-	-	D0
23	MULU A	8	1	$(A) \leftarrow (AL) \times (TL)$	-	-	dH	-	-	-	-	01
24	DIVU A	17	1	$(A) \leftarrow (T) / (A), \text{MOD} \rightarrow (T)$	dL	dH	dH	-	+	-	-	11
25	ANDW A	1	1	$(A) \leftarrow (A) \wedge (T)$	-	-	dH	+	+	R	-	63
26	ORW A	1	1	$(A) \leftarrow (A) \vee (T)$	-	-	dH	+	+	R	-	73
27	XORW A	1	1	$(A) \leftarrow (A) \vee (T)$	-	-	dH	+	+	R	-	53
28	CMP A	1	1	$(TL) - (AL)$	-	-	-	+	+	+	+	12
29	CMPW A	1	1	$(T) - (A)$	-	-	-	+	+	+	+	13
30	RORC A	1	1	$\text{ROR } C \rightarrow A$	-	-	-	+	+	-	+	0302
31	ROLCA	1	1	$\text{ROL } C \leftarrow A$	-	-	-	+	+	-	+	
32	CMP A, #d8	2	2	$(A) - d8$	-	-	-	+	+	+	+	14
33	CMP A, dir	3	2	$(A) - (dir)$	-	-	-	+	+	+	+	15
34	CMP A, @EP	2	1	$(A) - ((EP))$	-	-	-	+	+	+	+	17
35	CMP A, @IX + off	3	2	$(A) - ((IX) + off)$	-	-	-	+	+	+	+	16
36	CMP A, Ri	2	1	$(A) - (Ri)$	-	-	-	+	+	+	+	18 to 1F
37	DAA	1	1	decimaladjustforaddition	-	-	-	+	+	+	+	84
38	DAS	1	1	decimaladjustforsubtraction	-	-	-	+	+	+	+	94
39	XOR A	1	1	$(A) \leftarrow (AL) \vee (TL)$	-	-	-	+	+	R	-	52
40	XOR A, #d8	2	2	$(A) \leftarrow (AL) \vee d8$	-	-	-	+	+	R	-	54
41	XOR A, dir	3	2	$(A) \leftarrow (AL) \vee (dir)$	-	-	-	+	+	R	-	55
42	XOR A, @EP	2	1	$(A) \leftarrow (AL) \vee ((EP))$	-	-	-	+	+	R	-	57
43	XOR A, @IX + off	3	2	$(A) \leftarrow (AL) \vee ((IX) + off)$	-	-	-	+	+	R	-	56
44	XOR A, Ri	2	1	$(A) \leftarrow (AL) \vee (Ri)$	-	-	-	+	+	R	-	58 to 5F
45	AND A	1	1	$(A) \leftarrow (AL) \wedge (TL)$	-	-	-	+	+	R	-	62
46	AND A, #d8	2	2	$(A) \leftarrow (AL) \wedge d8$	-	-	-	+	+	R	-	64

Table E.4-2 Arithmetic Operation Instruction (2 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
47	AND A, dir	3	2	$(A) \leftarrow (AL) \wedge (\text{dir})$	-	-	-	+	+	R	-	65
48	AND A, @EP	2	1	$(A) \leftarrow (AL) \wedge ((EP))$	-	-	-	+	+	R	-	67
49	AND A, @IX + off	3	2	$(A) \leftarrow (AL) \wedge ((IX) + \text{off})$	-	-	-	+	+	R	-	66
50	AND A, Ri	2	1	$(A) \leftarrow (AL) \wedge (Ri)$	-	-	-	+	+	R	-	68 to 6F
51	OR A	1	1	$(A) \leftarrow (AL) \vee (TL)$	-	-	-	+	+	R	-	72
52	OR A, #d8	2	2	$(A) \leftarrow (AL) \vee d8$	-	-	-	+	+	R	-	74
53	OR A, dir	3	2	$(A) \leftarrow (AL) \vee (\text{dir})$	-	-	-	+	+	R	-	75
54	OR A, @EP	2	1	$(A) \leftarrow (AL) \vee ((EP))$	-	-	-	+	+	R	-	77
55	OR A, @IX + off	3	2	$(A) \leftarrow (AL) \vee ((IX) + \text{off})$	-	-	-	+	+	R	-	76
56	OR A, Ri	2	1	$(A) \leftarrow (AL) \vee (Ri)$	-	-	-	+	+	R	-	78 to 7F
57	CMP dir, #d8	4	3	$(\text{dir}) - d8$	-	-	-	+	+	+	+	95
58	CMP @EP, #d8	3	2	$((EP)) - d8$	-	-	-	+	+	+	+	97
59	CMP @IX + off, #d8	4	3	$((IX) + \text{off}) - d8$	-	-	-	+	+	+	+	96
60	CMP Ri, #d8	3	2	$(Ri) - d8$	-	-	-	+	+	+	+	98 to 9F
61	INCW SP	1	1	$(SP) \leftarrow (SP) + 1$	-	-	-	-	-	-	-	C1
62	DECW SP	1	1	$(SP) \leftarrow (SP) - 1$	-	-	-	-	-	-	-	D1

## ■ Branch Instructions

Table E.4-3 Branch Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	BZ/BEQ rel(at branch)	4	2	ifZ = 1 then PC ← PC + rel	-	-	-	-	-	-	-	FD
	BZ/BEQ rel(at no branch)	2										
2	BNZ/BNE rel(at branch)	4	2	ifZ = 0 then PC ← PC + rel	-	-	-	-	-	-	-	FC
	BNZ/BNE rel(at no branch)	2										
3	BC/BLO rel(at branch)	4	2	ifC = 1 then PC ← PC + rel	-	-	-	-	-	-	-	F9
	BC/BLO rel(at no branch)	2										
4	BNC/BHS rel(at branch)	4	2	ifC = 0 then PC ← PC + rel	-	-	-	-	-	-	-	F8
	BNC/BHS rel(at no branch)	2										
5	BN rel(at branch)	4	2	ifN = 1 then PC ← PC + rel	-	-	-	-	-	-	-	FB
	BN rel(at no branch)	2										
6	BP rel(at branch)	4	2	ifN = 0 then PC ← PC + rel	-	-	-	-	-	-	-	FA
	BP rel(at no branch)	2										
7	BLT rel(at branch)	4	2	ifV ∨ N = 1 then PC ← PC + rel	-	-	-	-	-	-	-	FF
	BLT rel(at no branch)	2										
8	BGE rel(at branch)	4	2	ifV ∨ N = 0 then PC ← PC + rel	-	-	-	-	-	-	-	FE
	BGE rel(at no branch)	2										
9	BBC dir : b, rel	5	3	if (dir : b) = 0 then PC ← PC + rel	-	-	-	-	+	-	-	B0 to B7
10	BBS dir : b, rel	5	3	if (dir : b) = 1 then PC ← PC + rel	-	-	-	-	+	-	-	B8 to BF
11	JMP @A	3	1	$(PC) \leftarrow (A)$	-	-	-	-	-	-	-	E0
12	JMP ext	4	3	$(PC) \leftarrow \text{ext}$	-	-	-	-	-	-	-	21
13	CALLV #vct	7	1	vectorcall	-	-	-	-	-	-	-	E8 to EF
14	CALL ext	6	3	subroutinecall	-	-	-	-	-	-	-	31
15	XCHW A, PC	3	1	$(PC) \leftarrow (A), (A) \leftarrow (PC) + 1$	-	-	dH	-	-	-	-	F4
16	RET	6	1	return from subroutine	-	-	-	-	-	-	-	20
17	RETI	8	1	return from interrupt	-	-	-	-	-	-	-	30

## ■ Other Instructions

Table E.4-4 Other Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OPCODE
1	PUSHW A	4	1	$((SP)) \leftarrow (A), (SP) \leftarrow (SP) - 2$	-	-	-	-	-	-	-	40
2	POPW A	3	1	$(A) \leftarrow ((SP)), (SP) \leftarrow (SP) + 2$	-	-	dH	-	-	-	-	50
3	PUSHW IX	4	1	$((SP)) \leftarrow (IX), (SP) \leftarrow (SP) - 2$	-	-	-	-	-	-	-	41
4	POPW IX	3	1	$(IX) \leftarrow ((SP)), (SP) \leftarrow (SP) + 2$	-	-	-	-	-	-	-	51
5	NOP	1	1	No operation	-	-	-	-	-	-	-	00
6	CLRC	1	1	$(C) \leftarrow 0$	-	-	-	-	-	-	R	81
7	SETC	1	1	$(C) \leftarrow 1$	-	-	-	-	-	-	S	91
8	CLRI	1	1	$(I) \leftarrow 0$	-	-	-	-	-	-	-	80
9	SETI	1	1	$(I) \leftarrow 1$	-	-	-	-	-	-	-	90

## MB95110B/M Series

## E.5 Instruction Map

Table E.5-1 shows the instruction map of F<sup>2</sup>MC-8FX.

### ■ Instruction Map

Table E.5-1 Instruction Map of F<sup>2</sup>MC-8FX

	H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
L																	
0	NOP	SWAP	RET	RETI	PUSHW	A	POPW	MOV	MOVW	CLRI	SETI	CLRB	BBC	INCW	DECW	JMP	MOVW
1	MULU	DIVU	JMP	CALL	PUSHW	IX	POPW	MOV	MOVW	CLRC	SETC	CLRB	BBC	INCW	DECW	MOVW	MOVW
2	ROLC	CMP	ADDC	SUBC	XCH	A, T	XOR	AND	OR	MOV	MOV	CLRB	BBC	INCW	DECW	MOVW	MOVW
3	RORC	CMPW	ADDCW	SUBCW	XCHW	A, T	XORW	ANDW	ORW	MOVW	MOVW	CLRB	BBC	INCW	DECW	MOVW	MOVW
4	MOV	CMP	ADDC	SUBC			XOR	AND	OR	DAA	DAS	CLRB	BBC	INCW	DECW	MOVW	MOVW
5	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
6	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
7	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
8	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
9	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
A	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
B	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
C	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
D	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
E	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW
F	MOV	CMP	ADDC	SUBC	MOV	MOV	XOR	AND	OR	MOV	CMP	CLRB	BBC	INCW	DECW	MOVW	MOVW

## APPENDIX F Mask Option

The mask option list of the MB95110B/M series is shown in Table F-1.

### ■ Mask Option List

Table F-1 Mask Option List 1

No.	Part number	MB95116B	MB95F118BS	MB95F118BW	MB95FV100D-101
	Specifying procedure	Specify when ordering MASK	Setting disabled	Setting disabled	Setting disabled
1	Clock mode select • Single clock product clock mode • Dual clock product clock mode	Selectable	Single clock product clock mode	Dual clock product clock mode	Changing by the switch on MCU board
2	Selection of oscillation stabilization wait time Selectable the initial value of main clock oscillation stabilization wait time.	Selectable 1 : $(2^2 - 2) / F_{CH}$ 2 : $(2^{12} - 2) / F_{CH}$ 3 : $(2^{13} - 2) / F_{CH}$ 4 : $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$

$F_{CH}$ : Main clock

Table F-2 Mask Option List 2 (1 / 2)

No.	Part number	MB95F114MS MB95F114NS MB95F114JS MB95F116MS MB95F116NS MB95F116JS MB95F116MAS MB95F116NAS	MB95F114MW MB95F114NW MB95F114JW MB95F116MW MB95F116NW MB95F116JW MB95F116MAW MB95F116NAW	MB95FV100D-103
	Specifying procedure	Setting disabled	Setting disabled	Setting disabled
1	Clock mode select • Single clock product clock mode • Dual clock product clock mode	Single clock product clock mode	Dual clock product clock mode	Changing by the switch on MCU board
2	Low voltage detection reset • With low voltage detection reset • Without low voltage detection reset	Specified by part number	Specified by part number	Changing by the switch on MCU board
3	Clock supervisor • With Clock supervisor • Without Clock supervisor	Specified by part number	Specified by part number	Changing by the switch on MCU board
4	Reset output • With reset output • Without reset output	Specified by part number	Specified by part number	If the clock supervisor is enabled by the switch on MCU board, the reset output is disabled. If the clock supervisor is disabled, the reset output is enabled.

**Table F-2 Mask Option List 2 (2 / 2)**

No.	Part number	MB95F114MS MB95F114NS MB95F114JS MB95F116MS MB95F116NS MB95F116JS MB95F116MAS MB95F116NAS	MB95F114MW MB95F114NW MB95F114JW MB95F116MW MB95F116NW MB95F116JW MB95F116MAW MB95F116NAW	MB95FV100D-103
	Specifying procedure	Setting disabled	Setting disabled	Setting disabled
5	Oscillation stabilization wait time	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$

$F_{CH}$ : Main clock

**Table F-3 Mask Option List 3**

No.	Part number	MB95117M	MB95F118MS MB95F118NS MB95F118JS	MB95F118MW MB95F118NW MB95F118JW	MB95FV100D-103
	Specifying procedure	Setting disabled	Setting disabled	Setting disabled	Setting disabled
1	Clock mode select • Single clock product clock mode • Dual clock product clock mode	Specify when ordering MASK	Single clock product clock mode	Dual clock product clock mode	Changing by the switch on MCU board
2	Low voltage detection reset • With low voltage detection reset • Without low voltage detection reset	Specify when ordering MASK	Specified by part number	Specified by part number	Changing by the switch on MCU board
3	Clock supervisor • With Clock supervisor • Without Clock supervisor	Specify when ordering MASK	Specified by part number	Specified by part number	Changing by the switch on MCU board
4	Reset output • With reset output • Without reset output	Specify when ordering MASK	Specified by part number	Specified by part number	If the clock supervisor is enabled by the switch on MCU board, the reset output is disabled. If the clock supervisor is disabled, the reset output is enabled.
5	Oscillation stabilization wait time	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$	Fixed to oscillation stabilization wait time of $(2^{14} - 2) / F_{CH}$

$F_{CH}$ : Main clock



## **APPENDIX G Writing to Flash Microcontroller Using Parallel Writer**

---

**This section describes writing to flash microcontroller using parallel writer.**

---

### **■ Writing to Flash Microcontroller Using Parallel Writer**

**Table G-1 Parallel Writer and Adaptor**

Package	Compatible adaptor model	Parallel writer
	Flash Support Group, Inc.	
FPT-52P-M01	developing	AF9708 (Ver 02.35G higher)
FPT-48P-M26	TEF110-118F37AP	AF9709/B (Ver 02.35G higher)
LCC-48P-M09	TEF110-118F41AP	AF9723+AF9834 (Ver 02.08E higher)

Contact: Flash Support Group, Inc. Tel: +81-53-428-8380

## ■ Sector Configuration

The following diagram shows the addresses corresponding to each sector on accessing using CPU and parallel writer.

**Figure G-1 CPU Address and Writer Address**

- Products other than MB95F116MAW/F116NAW/F116MAS/F116NAS

Flash memory	CPU address	Writer address*
SA1 (4KB)	1000 <sub>H</sub>	71000 <sub>H</sub>
	1FFF <sub>H</sub>	71FFF <sub>H</sub>
SA2 (4KB)	2000 <sub>H</sub>	72000 <sub>H</sub>
	2FFF <sub>H</sub>	72FFF <sub>H</sub>
SA3 (4KB)	3000 <sub>H</sub>	73000 <sub>H</sub>
	3FFF <sub>H</sub>	73FFF <sub>H</sub>
SA4 (16KB)	4000 <sub>H</sub>	74000 <sub>H</sub>
	7FFF <sub>H</sub>	77FFF <sub>H</sub>
SA5 (16KB)	8000 <sub>H</sub>	78000 <sub>H</sub>
	BFFF <sub>H</sub>	7BFFF <sub>H</sub>
SA6 (4KB)	C000 <sub>H</sub>	7C000 <sub>H</sub>
	CFFF <sub>H</sub>	7CFFF <sub>H</sub>
SA7 (4KB)	D000 <sub>H</sub>	7D000 <sub>H</sub>
	DFFF <sub>H</sub>	7DFFF <sub>H</sub>
SA8 (4KB)	E000 <sub>H</sub>	7E000 <sub>H</sub>
	FFFF <sub>H</sub>	7EFFF <sub>H</sub>
SA9 (4KB)	F000 <sub>H</sub>	7F000 <sub>H</sub>
	FFFF <sub>H</sub>	7FFFF <sub>H</sub>

Lower bank

Upper bank

- MB95F116MAW/F116NAW/F116MAS/F116NAS

Flash memory	CPU address	Writer address*
32 Kbytes	8000 <sub>H</sub>	78000 <sub>H</sub>
	FFFF <sub>H</sub>	7FFFF <sub>H</sub>

\*: The writer address is equivalent to the CPU address when data is written to the flash memory using parallel writer.

When a parallel writer is used for writing/erasing, the writer address is used for writing/erasing.

## ■ How to Write

- For 3V products, set the type code of parallel writer to "17226".  
For 5V products, set the type code of parallel writer to "17222".
- Load program data to 71000<sub>H</sub> to 7FFFF<sub>H</sub> of parallel writer.
- Write using parallel writer.



# INDEX

---

The index follows on the next page.  
This is listed in alphabetic order.

---

# Index

## Numerics

### 16-bit

16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0) ....	293
16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0) ...	292
16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) ...	294
16-bit PPG Status Control Register, Lower (PCNTL0) .....	297
16-bit PPG Status Control Register, Upper (PCNTH0) .....	295
16-bit PPG Timer .....	286
Block Diagram of 16-bit PPG Timer .....	287
Block Diagrams of Pins Related to 16-bit PPG .....	290
Channels of 16-bit PPG Timer .....	289
Interrupts of 16-bit PPG Timer .....	299
Notes on Using 16-bit PPG Timer .....	304
Operation of 16-bit PPG Mode .....	280
Pins of 16-bit PPG Timer .....	290
Placement of 16-bit Data in Memory .....	47
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers of 16-bit PPG Timer .....	291
Sample Programs for 16-bit PPG Timer .....	305
Setting 16-bit PPG Mode .....	279

### 16-bit PPG Cycle Setting Buffer Registers

16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0) ....	293
---	-----

### 16-bit PPG Down Counter Registers

16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0) ...	292
--	-----

### 16-bit PPG Duty Setting Buffer Registers

16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) ...	294
---	-----

### 16-bit PPG Status Control Register

16-bit PPG Status Control Register, Lower (PCNTL0) .....	297
16-bit PPG Status Control Register, Upper (PCNTH0) .....	295

### 256-Kbit

Features of 256-Kbit Flash Memory .....	564
Overview of 256-Kbit Flash Memory .....	564
Sector Configuration of 256-Kbit Flash Memory .....	565

### 8/10-bit

Block Diagram of 8/10-bit A/D Converter .....	491
Block Diagram of Pins Related to 8/10-bit A/D Converter .....	494

### Interrupts During 8/10-bit A/D Converter Operation

.....	501
-------	-----

### List of 8/10-bit A/D Converter Registers .....

.....	495
-------	-----

### Notes on Use of 8/10-bit A/D Converter .....

.....	505
-------	-----

### Operations of 8/10-bit A/D Converter's Conversion

Function.....	502
---------------	-----

### Pins of 8/10-bit A/D Converter.....

.....	493
-------	-----

### Register and Vector Table Related to 8/10-bit A/D

Converter Interrupts .....	501
----------------------------	-----

### Sample Programs for 8/10-bit A/D Converter .....

.....	506
-------	-----

### 8/10-bit A/D Converter Control Register

8/10-bit A/D Converter Control Register 1 (ADC1) .....	496
---	-----

8/10-bit A/D Converter Control Register 2 (ADC2) .....	498
---	-----

### 8/10-bit A/D Converter Data Registers

8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL) .....	500
---	-----

### 8/16-bit

8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0).....	225
---	-----

8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1).....	228
---	-----

8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR) .....	234
--	-----

8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) .....	231
---	-----

8/16-bit PPG Output Inversion Register (REVC) .....	272
--	-----

8/16-bit PPG Start Register (PPGS) .....	271
--	-----

8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
---	-----

8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00).....	269
---	-----

8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00) .....	270
---	-----

8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
---	-----

Block Diagram of 8/16-bit Compound Timer.....	219
--	-----

Block Diagram of 8/16-bit PPG.....	259
------------------------------------	-----

Block Diagram of Pins Related to 8/16-bit Compound Timer.....	223
--	-----

Block Diagram of Pins Related to 8/16-bit PPG .....	263
--	-----

Channels of 8/16-bit Compound Timer .....	221
---	-----

Channels of 8/16-bit PPG .....	261
--------------------------------	-----

Interrupts of 8/16-bit PPG.....	273
---------------------------------	-----

LIN Synch Field Edge Detection Interrupt (8/16-bit Compound Timer Interrupt)....	402
---	-----

Notes on Using 8/16-bit Compound Timer .....	256
Notes on Using 8/16-bit PPG .....	281
Overview of 8/16-bit PPG.....	258
Pins of 8/16-bit PPG.....	262
Pins Related to 8/16-bit Compound Timer .....	222
Registers and Vector Table Related to Interrupts of 8/16-bit PPG.....	273
Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer.....	238
Registers of 8/16-bit PPG .....	264
Registers Related to 8/16-bit Compound Timer .....	224
Sample Programs for 8/16-bit PPG Timer .....	282
<b>8/16-bit Compound Timer 00/01 Control Status Register</b>	
8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0).....	225
8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1).....	228
<b>8/16-bit Compound Timer 00/01 Data Register</b>	
8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR).....	234
<b>8/16-bit Compound Timer 00/01 Timer Mode Control Register</b>	
8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCR0) .....	231
<b>8/16-bit PPG Output Inversion Register</b>	
8/16-bit PPG Output Inversion Register (REVC).....	272
<b>8/16-bit PPG Start Register</b>	
8/16-bit PPG Start Register (PPGS) .....	271
<b>8/16-bit PPG Timer 00 Control Register</b>	
8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
<b>8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register</b>	
8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00).....	269
<b>8/16-bit PPG Timer 00/01 Duty Setup Buffer Register</b>	
8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00).....	270
<b>8/16-bit PPG Timer 01 Control Register</b>	
8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
<b>8-bit</b>	
Operation of 8-bit PPG Independent Mode .....	275
Operation of 8-bit Prescaler + 8-bit PPG Mode .....	277
Setting 8-bit Independent Mode.....	275
Setting 8-bit Prescaler + 8-bit PPG Mode .....	277
<b>8FX</b>	
Notes on F <sup>2</sup> MC-8FX Software Development Support Environment (MB95FV100D and MB2146- 09) .....	562

## A

<b>A/D Conversion</b>	
A/D Conversion Functions .....	490
Operations of A/D Conversion Function .....	503
<b>A/D Converter</b>	
Block Diagram of 8/10-bit A/D Converter.....	491
Block Diagram of Pins Related to 8/10-bit A/D Converter Block Diagram.....	494
Interrupts During 8/10-bit A/D Converter Operation .....	501
List of 8/10-bit A/D Converter Registers.....	495
Notes on Use of 8/10-bit A/D Converter .....	505
Operations of 8/10-bit A/D Converter's Conversion Function .....	502
Pins of 8/10-bit A/D Converter .....	493
Register and Vector Table Related to 8/10-bit A/D Converter Interrupts .....	501
Sample Programs for 8/10-bit A/D Converter .....	506
<b>Access Sector Map</b>	
Access Sector Map Based on Sector Conversion Enable Bit (FSR:SSN) .....	594
<b>Acknowledgment</b>	
Acknowledgment.....	476
<b>ADC</b>	
8/10-bit A/D Converter Control Register 1 (ADC1) .....	496
8/10-bit A/D Converter Control Register 2 (ADC2) .....	498
<b>ADDH</b>	
8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL).....	500
<b>ADDL</b>	
8/10-bit A/D Converter Data Registers Upper/Lower (ADDH, ADDL).....	500
<b>Address Register</b>	
I <sup>2</sup> C Address Register (IAAR0).....	467
<b>Addressing</b>	
Addressing.....	475
Explanation of Addressing.....	613
<b>Applicable Addresses</b>	
Wild Register Applicable Addresses.....	213
<b>Arbitration</b>	
Arbitration .....	478
<b>Arithmetic Operation</b>	
Arithmetic Operation Instructions .....	623
<b>Asynchronous LIN Mode</b>	
Asynchronous LIN Mode Operation .....	426
<b>Asynchronous Mode</b>	
Asynchronous Mode Operation.....	418

## INDEX

<b>B</b>	
<b>Banks</b>	
Interrupt occurrence when programming upper bank .....	559
Sector/Bank Configuration of Dual-Operation Flash Memory .....	530
<b>Baud rate</b>	
Baud Rate Calculation .....	410
Baud Rate Setting .....	375
LIN-UART Baud Rate Selection .....	408
Reload Value and Baud Rate of Each Clock Speed .....	411
<b>Baud Rate Generator</b>	
Block Diagram of UART/SIO Dedicated Baud Rate Generator .....	370
Channels of UART/SIO Dedicated Baud Rate Generator .....	371
Registers Related to UART/SIO Dedicated Baud Rate Generator .....	372
<b>BGR</b>	
Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0) .....	399
<b>Bi-directional Communication</b>	
Bi-directional Communication Function .....	430
<b>Bit 5</b>	
One-shot Mode (MDSE of PCNTH0 Register: bit 5=1) .....	302
PWM Mode (MDSE of PCNTH Register: bit 5=0) .....	300
<b>Bit Manipulation Instructions</b>	
Read Destination on the Execution of Bit Manipulation Instructions .....	621
<b>Bits Result</b>	
Bits Result Information Bits .....	43
<b>Block Diagram</b>	
Block Diagram of 16-bit PPG Timer .....	287
Block Diagram of 8/10-bit A/D Converter .....	491
Block Diagram of 8/16-bit Compound Timer .....	219
Block Diagram of 8/16-bit PPG .....	259
Block Diagram of All MB95110B/M Series .....	10
Block Diagram of Clock Supervisor .....	517
Block Diagram of External Interrupt Circuit .....	311
Block Diagram of Interrupt Pin Selection Circuit .....	325
Block Diagram of LIN-UART Pins .....	385
Block Diagram of Low-voltage Detection Reset Circuit .....	511
Block Diagram of Pins Related to 8/10-bit A/D Converter Block Diagram .....	494
Block Diagram of Pins Related to 8/16-bit Compound Timer .....	223
Block Diagram of Pins Related to 8/16-bit PPG .....	263
Block Diagram of Pins Related to External Interrupt Circuit .....	313
Block Diagram of Pins Related to UART/SIO .....	339
Block Diagram of Port 0 .....	116
Block Diagram of Port 1 .....	121
Block Diagram of Port 2 .....	126
Block Diagram of Port 3 .....	131
Block Diagram of Port 5 .....	136
Block Diagram of Port 6 .....	141
Block Diagram of Port G .....	146
Block Diagram of the Clock Controller .....	51
Block Diagram of Time-base Timer .....	153
Block Diagram of UART/SIO .....	335
Block Diagram of UART/SIO Dedicated Baud Rate Generator .....	370
Block Diagram of Watch Counter .....	191
Block Diagram of Watch Prescaler .....	177
Block Diagram of Watchdog Timer .....	167
Block Diagram of Wild Register Function .....	205
Block Diagrams of Pins Related to 16-bit PPG .....	290
I <sup>2</sup> C Block Diagram .....	452
I <sup>2</sup> C-related Pin Block Diagram .....	456
LIN-UART Block Diagram .....	381
Prescaler Block Diagram .....	87
<b>Branch</b>	
Branch Instructions .....	624
<b>BRSR</b>	
UART/SIO Dedicated Baud Rate Generator Baud Rate Setting Register (BRSR0) ....	374
<b>Bus Control Register</b>	
I <sup>2</sup> C Bus Control Register 0 (IBCR00) .....	458
I <sup>2</sup> C Bus Control Register 1 (IBCR10) .....	461
<b>Bus Interface</b>	
Pins Related to I <sup>2</sup> C Bus Interface .....	455
<b>Bus Status Register</b>	
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
<b>C</b>	
<b>Causes</b>	
Table of Interrupt Causes .....	605
<b>CCR</b>	
Condition Code Register (CCR) Configuration .....	43
<b>Channel</b>	
Channels of UART/SIO .....	337
Channels of UART/SIO Dedicated Baud Rate Generator .....	371
<b>Channels</b>	
Channels of 16-bit PPG Timer .....	289
Channels of 8/16-bit Compound Timer .....	221
Channels of 8/16-bit PPG .....	261
Channels of External Interrupt Circuit .....	312
I <sup>2</sup> C Channels .....	454
<b>Check</b>	
Check That Clock-mode Transition has been Completed before Setting Standby Mode. 76	

Chip erase	
Erasing All Data from Flash Memory	
(Chip Erase) .....	554, 578
Notes on Chip Erase .....	554, 578
Clock	
Block Diagram of the Clock Controller .....	51
Check That Clock-mode Transition has been	
Completed before Setting Standby Mode. 76	
Clock Mode State Transition Diagram .....	70
Clock Oscillator Circuit .....	84
Combinations of Clock Mode and Standby Mode..	55
Configuration of System Clock Control Register	
(SYCC).....	58
External Clock .....	412
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
Input Clock .....	87, 220, 260, 288, 453
Input clock	
.. 154, 168, 178, 192, 336, 370, 384, 492	
Operations in Main Clock Mode.....	69
Operations in Main PLL Clock Mode .....	69
Operations in Sub PLL Clock Mode	
(on Dual Clock Product) .....	69
Oscillation Stabilization Wait Time and Clock Mode/	
Standby Mode Transition.....	57
Output Clock .....	87, 154, 178, 370
Overview of Clock Controller .....	50
Peripheral Resources not Affected	
by Clock Mode .....	53
PLL Clock Oscillation Stabilization Wait Time ....	57
Reload Value and Baud Rate of Each Clock Speed	
.....	411
Clock Control Register	
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
Clock Controller	
Block Diagram of the Clock Controller .....	51
Overview of Clock Controller .....	50
Clock Mode	
Clock Mode State Transition Diagram .....	70
Clock Modes .....	53
Combinations of Clock Mode and Standby	
Mode .....	55
Operations in Main Clock Mode.....	69
Operations in Main PLL Clock Mode .....	69
Operations in Sub PLL Clock Mode	
(on Dual Clock Product) .....	69
Oscillation Stabilization Wait Time and Clock Mode/	
Standby Mode Transition.....	57
Peripheral Resources not Affected	
by Clock Mode .....	53
Clock Speed	
Reload Value and Baud Rate of Each Clock Speed	
.....	411
Clock supervisor	
Block Diagram of Clock Supervisor.....	517
Example Operation Flowchart for the Clock	
Supervisor .....	523
Example Startup Flowchart when using the Clock	
Supervisor.....	524
Notes on using the Clock Supervisor .....	525
Operations of Clock Supervisor .....	522
Overview of Clock Supervisor .....	516
Register of Clock Supervisor .....	519
Clock supervisor control register	
Clock supervisor control register	
(CSVCR) .....	520
Combinations	
Combinations of Clock Mode and Standby Mode ..	55
Command	
Command Sequence Table .....	540, 569
Notes on Issuing Commands.....	541, 569
Compound Timer	
8/16-bit Compound Timer 00/01 Control Status	
Register 0 (T00CR0/T01CR0) .....	225
8/16-bit Compound Timer 00/01 Control Status	
Register 1 (T00CR1/T01CR1) .....	228
8/16-bit Compound Timer 00/01 Data Register	
(T00DR/T01DR) .....	234
8/16-bit Compound Timer 00/01 Timer Mode Control	
Register ch.0 (TMCRO) .....	231
Block Diagram of 8/16-bit Compound Timer .....	219
Block Diagram of Pins Related to 8/16-bit Compound	
Timer .....	223
Channels of 8/16-bit Compound Timer .....	221
Notes on Using 8/16-bit Compound Timer .....	256
Pins Related to 8/16-bit Compound Timer .....	222
Registers and Vector Tables Related to Interrupts	
of 8/16-bit Compound Timer .....	238
Registers Related to 8/16-bit Compound Timer ...	224
Compound Timer Interrupt	
LIN Synch Field Edge Detection Interrupt	
(8/16-bit Compound Timer Interrupt) ....	402
Condition	
Start Conditions.....	475
Condition Code Register	
Condition Code Register (CCR) Configuration.....	43
Configuration	
Configuration of Direct Bank Pointer (DP).....	41
Configuration of Memory Space .....	30
Configuration of Oscillation Stabilization Wait Time	
Setting Register (WATR) .....	63
Configuration of PLL Control Register	
(PLLC) .....	60
Configuration of System Clock Control Register	
(SYCC) .....	58
Port 0 Configuration .....	115
Port 1 Configuration .....	120
Port 2 Configuration .....	125
Port 3 Configuration .....	130
Port 5 Configuration .....	135
Port 6 Configuration .....	140
Port G Configuration.....	145



## INDEX

Register Bank Pointer, Direct Bank Pointer Mirror Address .....	40
Sector Configuration.....	629
Continuous Mode	
Interval Timer Function (Continuous Mode) .....	216
Operation of Interval Timer Function (Continuous Mode).....	241
Control Register	
16-bit PPG Status Control Register, Lower (PCNTL0) .....	297
16-bit PPG Status Control Register, Upper (PCNTH0).....	295
8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
Conversion	
Access Sector Map Based on Sector Conversion Enable Bit (FSR:SSEN) .....	594
Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN) .....	595
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN) .....	593
Counter	
16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0) .....	292
Function of Reload Counter.....	414
Watch counter.....	190
CPU	
Inter-CPU Connection Method .....	417
Standby Mode is Also Canceled when the CPU Rejects Interrupts. ....	76
CSVCR	
Clock supervisor control register (CSVCR).....	520
<b>D</b>	
Data Polling Flag	
Data Polling Flag (DQ7) .....	544, 571
Data Register	
I <sup>2</sup> C Data Register (IDDR0) .....	466
Data Transfer	
Data Transfer .....	475
Debug	
Precautions for Debug .....	25
Dedicated Baud Rate Generator	
Operation of Dedicated Baud Rate Generator (Reload Counter) .....	413
Dedicated Registers	
Configuration of Dedicated Registers .....	38
Functions of Dedicated Registers .....	38
Description	
Pin Description .....	17

Device	
Handling Devices.....	24
Difference Points among Products	
Difference Points among Products and Notes on Selecting a Product .....	8
Display Sign	
Explanation of Display Sign of Instruction .....	611
DP	
Configuration of Direct Bank Pointer (DP).....	41
DQ3	
Sector Erase Timer Flag (DQ3) .....	549
DQ5	
Timing Limit Elapsed Flag (DQ5) .....	548, 573
DQ6	
Restrictions on Toggle Bit Flag (DQ6).....	562
Toggle Bit Flag (DQ6) .....	546, 572
DQ7	
Data Polling Flag (DQ7).....	544, 571
Dual Clock Product	
Operations in Sub Clock Mode (on Dual Clock Product) .....	69
Operations in Sub PLL Clock Mode (on Dual Clock Product) .....	69
Dual Operation Flash	
Features of Dual Operation Flash .....	592
Dual-Operation	
Features of Dual-Operation Flash Memory .....	529
Overview of Dual-Operation Flash Memory .....	528
Sector/Bank Configuration of Dual-Operation Flash Memory .....	530
<b>E</b>	
ECCR	
Bit Configuration of LIN-UART Extended Communication Control Register (ECCR) .....	397
Edge Detection Interrupt	
LIN Synch Field Edge Detection Interrupt (8/16-bit Compound Timer Interrupt)....	402
EIC	
External Interrupt Control Register (EIC00) .....	315
Enable Transmission/Reception	
Enable Transmission/Reception.....	417
Erasing	
Details of Programming/Erasing Flash Memory .....	550, 574
Erasing All Data from Flash Memory (Chip Erase) .....	554, 578
Erasing Arbitrary Data from Flash Memory (Sector Erasing) .....	555
Flash memory program/erase .....	564
Flash Memory Programming/Erasing.....	529
Notes on Chip Erase.....	554, 578
Notes on Erasing Data from Sectors.....	555

Operation During Programming/Erasing .....	596
Programming/Erasing Operation .....	560
Resuming Sector Erasing from Flash Memory ....	558
Suspending Sector Erasing from Flash Memory ..	557
<b>Erasing Procedure</b>	
Flash Memory Sector Erasing Procedure .....	555
<b>ESCR</b>	
Bit Configuration of LIN-UART Extended Status Control Register (ESCR) .....	395
<b>Example</b>	
Setup Procedure Example .....	274, 303, 319, 473
<b>Explanation</b>	
Explanation of Addressing .....	613
Explanation of Display Sign of Instruction .....	611
Explanation of Item in Instruction Table .....	612
<b>External Clock</b>	
External Clock .....	412
<b>External Interrupt</b>	
Block Diagram of External Interrupt Circuit .....	311
Block Diagram of Pins Related to External Interrupt Circuit.....	313
Channels of External Interrupt Circuit.....	312
External Interrupt Control Register (EIC00) .....	315
Functions of External Interrupt Circuit .....	310
Interrupt During Operation of External Interrupt Circuit.....	317
List of Registers of External Interrupt Circuit .....	314
Notes on Using External Interrupt Circuit.....	320
Operation of External Interrupt Circuit.....	318
Pins Related to External Interrupt Circuit .....	313
Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Sample Programs for External Interrupt Circuit ..	321
<b>External Interrupt Circuit</b>	
Block Diagram of External Interrupt Circuit.....	311
Block Diagram of Pins Related to External Interrupt Circuit.....	313
Channels of External Interrupt Circuit.....	312
Functions of External Interrupt Circuit .....	310
Interrupt During Operation of External Interrupt Circuit.....	317
List of Registers of External Interrupt Circuit .....	314
Notes on Using External Interrupt Circuit.....	320
Operation of External Interrupt Circuit.....	318
Pins Related to External Interrupt Circuit .....	313
Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Sample Programs for External Interrupt Circuit ..	321
<b>External Interrupt Control Register</b>	
External Interrupt Control Register (EIC00) .....	315
<b>F</b>	
<b>F<sup>2</sup>MC-8FX</b>	
Instruction Overview of F <sup>2</sup> MC-8FX .....	610
Notes on Software Programming Support Environment of F <sup>2</sup> MC-8FX (MB95FV100D and MB2146-09) .....	597
<b>Feature</b>	
Feature of MB95110B/M Series.....	2
Features of Dual Operation Flash .....	592
<b>Fixed-cycle Mode</b>	
Operation of PWM Timer Function (Fixed-cycle Mode).....	245
PWM Timer Function (Fixed-cycle Mode).....	216
<b>Flag Set</b>	
Reception Interrupt Generation and Flag Set Timing .....	404
Transmit Interrupt Generation and Flag Set Timing .....	406
<b>Flash</b>	
Features of Dual Operation Flash .....	592
<b>Flash Memory</b>	
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN).....	593
<b>Flash memory</b>	
Details of Programming/Erasing Flash Memory .....	550, 574
Erasing All Data from Flash Memory (Chip Erase).....	554, 578
Erasing Arbitrary Data from Flash Memory (Sector Erasing).....	555
Features of 256-Kbit Flash Memory .....	564
Features of Dual-Operation Flash Memory.....	529
Flash Memory Programming Procedure .....	552, 576
Flash Memory Programming/Erasing .....	529, 564
Flash Memory Sector Erasing Procedure.....	555
Overview of 256-Kbit Flash Memory .....	564
Overview of Dual-Operation Flash Memory.....	528
Placing Flash Memory in the Read/Reset State .....	551, 575
Programming Data into Flash Memory .....	552
Programming Data into Flash Memory Write .....	576
Register of Flash Memory .....	531, 566
Resuming Sector Erasing from Flash Memory .....	558
Sector Configuration of 256-Kbit Flash Memory .....	565
Sector/Bank Configuration of Dual-Operation Flash Memory .....	530
Suspending Sector Erasing from Flash Memory...	557
<b>Flash memory products</b>	
Basic Configuration of Serial Programming Connection for Flash Memory Products .....	582
<b>Flash Memory Sector Write Control Register</b>	
Flash Memory Sector Write Control Registers (SWRE0/SWRE1) .....	535
Flash Memory Sector Write Control Register (SWRE0/SWRE1) Setup Flowchart.....	538
<b>Flash Memory Status Register</b>	
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN).....	593

## INDEX

Flash memory status register	
Flash memory status register (FSR).....	532, 567
Flash Microcontroller	
Writing to Flash Microcontroller	
Using Parallel Writer .....	628
Flash Microcontroller Programmer	
Example of Minimum Connection to Flash	
Microcontroller Programmer .....	588
Flash Security	
Flash Security .....	561, 579
FPT-48P-M26	
Package Dimension of FPT-48P-M26 .....	15
FPT-52P-M01	
Package Dimension of FPT-52P-M01 .....	14
Free-run Mode	
Interval Timer Function (Free-run Mode) .....	216
Operation of Interval Timer Function	
(Free-run Mode).....	243
FSR	
Access Sector Map Based on Sector Conversion	
Enable Bit (FSR:SSEN) .....	594
Flash memory status register (FSR).....	532, 567
Note on Setting the FSR:WRE.....	539
Procedure of Setting the Sector Conversion Enable Bit	
(FSR:SSEN) .....	595
Sector Conversion Enable Bit in Flash Memory Status	
Register (FSR:SSEN) .....	593
Setup Procedure of Sector Conversion Enable Bit	
(FSR:SSEN) Setup Procedure.....	559
Function	
Function to Wake Up the MCU	
from Standby Mode .....	481
Functions of External Interrupt Circuit .....	310
I <sup>2</sup> C Functions.....	450
Input Capture Function .....	217
Interval Timer Function	
(Continuous Mode).....	216
Interval Timer Function (Free-run Mode) .....	216
Interval Timer Function (One-shot Mode) .....	216
Operation of Input Capture Function .....	251
Operation of Interval Timer Function	
(Continuous Mode).....	241
Operation of Interval Timer Function	
(Free-run Mode).....	243
Operation of Interval Timer Function	
(One-shot Mode) .....	239
Operation of PWC Timer Function .....	249
Operation of PWM Timer Function	
(Fixed-cycle Mode) .....	245
Operation of PWM Timer Function	
(Variable-cycle Mode) .....	247
Port 0 Register Function.....	117
Port 1 Register Function.....	122
Port 2 Register Function.....	127
Port 3 Register Function.....	132

Port 5 Register Function .....	137
Port 6 Register Function .....	142
Port G Register Function .....	147
PWC Timer Function .....	217
PWM Timer Function (Fixed-cycle Mode).....	216
PWM Timer Function	
(Variable-cycle Mode).....	216
When Interval Timer, Input Capture, or PWC Function	
Has Been Selected.....	254

## G

General Call Address	
General Call Address .....	477
General-purpose Register	
Configuration of General-purpose Registers .....	45
Features of General-purpose Registers .....	46
General-purpose Register Area	
(Addresses: 0100 <sub>H</sub> to 01FF <sub>H</sub> ).....	32

## H

Hardware	
Hardware Connection Example .....	214
Hardware sequence flag	
Hardware sequence flag.....	542, 570
Hardware Trigger	
Hardware Trigger.....	303
How to Write	
How to Write.....	629

## I

I/O Circuit	
I/O Circuit Type .....	20
I/O Map	
I/O Map .....	600
I/O Ports	
Overview of I/O Ports .....	114
I <sup>2</sup> C	
I <sup>2</sup> C Address Register (IAAR0) .....	467
I <sup>2</sup> C Block Diagram .....	452
I <sup>2</sup> C Bus Control Register 0 (IBCR00) .....	458
I <sup>2</sup> C Bus Control Register 1 (IBCR10) .....	461
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
I <sup>2</sup> C Channels .....	454
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
I <sup>2</sup> C Data Register (IDDR0).....	466
I <sup>2</sup> C Functions .....	450
I <sup>2</sup> C Protocol .....	474
I <sup>2</sup> C Registers .....	457
I <sup>2</sup> C Sample Programs .....	485
I <sup>2</sup> C System .....	474
I <sup>2</sup> C-related Pin Block Diagram.....	456
Notes on Use of I <sup>2</sup> C .....	483
Operation of I <sup>2</sup> C .....	473
Pins Related to I <sup>2</sup> C Bus Interface.....	455

Registers and Vector Table Related to I <sup>2</sup> C	
Interrupts.....	472
I <sup>2</sup> C Address Register	
I <sup>2</sup> C Address Register (IAAR0) .....	467
I <sup>2</sup> C Bus Control Register	
I <sup>2</sup> C Bus Control Register 0 (IBCR00) .....	458
I <sup>2</sup> C Bus Control Register 1 (IBCR10) .....	461
I <sup>2</sup> C Bus Status Register	
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
I <sup>2</sup> C Clock Control Register	
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
I <sup>2</sup> C Data Register	
I <sup>2</sup> C Data Register (IDDR0) .....	466
I <sup>2</sup> C Registers	
I <sup>2</sup> C Registers .....	457
IAAR	
I <sup>2</sup> C Address Register (IAAR0) .....	467
IBCR	
I <sup>2</sup> C Bus Control Register 0 (IBCR00) .....	458
I <sup>2</sup> C Bus Control Register 1 (IBCR10) .....	461
IBSR	
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
ICCR	
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
IDDR	
I <sup>2</sup> C Data Register (IDDR0) .....	466
ILR	
Interrupt Level Setting Registers (ILR0 to ILR5)	
Configuration .....	104
Input	
Input Capture Function .....	217
Input Clock .....	87, 220, 260, 288, 453
Operation of Input Capture Function .....	251
When Interval Timer, Input Capture, or PWC Function	
Has Been Selected.....	254
Input Capture	
Input Capture Function .....	217
Operation of Input Capture Function .....	251
When Interval Timer, Input Capture, or PWC Function	
Has Been Selected.....	254
Input clock	
Input clock	
.. 154, 168, 178, 192, 336, 370, 384, 492	
Instruction	
Arithmetic Operation Instructions .....	623
Branch Instructions .....	624
Explanation of Display Sign of Instruction .....	611
Explanation of Item in Instruction Table .....	612
Instruction Map .....	625
Instruction Overview of F <sup>2</sup> MC-8FX .....	610
Other Instructions .....	624
Place at Least Three NOP Instructions Immediately	
Following a Standby Mode Setting	
Instruction. ....	76
Read Destination on the Execution of Bit	
Manipulation Instructions .....	621
Special Instruction .....	617
Transfer Instructions .....	622
Instruction Map	
Instruction Map .....	625
Inter-CPU Connection Method	
Inter-CPU Connection Method.....	417
Interface	
Pins Related to I <sup>2</sup> C Bus Interface .....	455
Interrupt	
An Interrupt Request may Suppress Transition to	
Standby Mode. ....	76
Block Diagram of External Interrupt Circuit.....	311
Block Diagram of Pins Related to External Interrupt	
Circuit .....	313
Channels of External Interrupt Circuit .....	312
External Interrupt Control Register (EIC00) .....	315
Functions of External Interrupt Circuit.....	310
Interrupt Acceptance Control Bits .....	44
Interrupt During Operation of External Interrupt	
Circuit .....	317
Interrupt Generated when Upper Banks are	
Updated .....	595
Interrupt occurrence when programming upper bank	
.....	559
Interrupt Processing Stack Area .....	111
Interrupt Processing Steps .....	105
Interrupt Processing Time.....	109
Interrupt when Interval Function is in Operation..	158
Interrupt when Interval Timer Function is in Operation	
(Watch Interrupts).....	182
Interrupts During 8/10-bit A/D Converter Operation	
.....	501
Interrupts of 16-bit PPG Timer.....	299
Interrupts of 8/16-bit PPG .....	273
Interrupts of UART/SIO.....	349
Interrupts of Watch Counter .....	197
Interrupts of Watch Prescaler .....	182
LIN Synch Field Edge Detection Interrupt	
(8/16-bit Compound Timer Interrupt) ....	402
List of Registers of External Interrupt Circuit .....	314
Nested Interrupts .....	108
Notes on Using External Interrupt Circuit .....	320
Operation of External Interrupt Circuit .....	318
Overview of Interrupts .....	102
Pins Related to External Interrupt Circuit.....	313
Reception interrupt .....	349, 400
Reception Interrupt Generation and Flag Set Timing	
.....	404
Register and Vector Table for Interrupts of Time-base	
Timer .....	159
Register and Vector Table Related to 8/10-bit A/D	
Converter Interrupts .....	501
Register and Vector Table Related to Interrupts of	
Watch Counter .....	197

## INDEX

Register and Vector Table Related to Interrupts of Watch Prescaler .....	183
Register and Vector Table Related to LIN-UART Interrupt .....	403
Registers and Vector Table Related to I <sup>2</sup> C Interrupts .....	472
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers and Vector Table Related to Interrupts of 8/16-bit PPG .....	273
Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Registers and Vector Table Related to UART/SIO Interrupts .....	349
Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer .....	238
Sample Programs for External Interrupt Circuit .....	321
Stack Operation at Start of Interrupt Processing ..	110
Stack Operation upon Returning from Interrupt...	110
Standby Mode is Also Canceled when the CPU Rejects Interrupts. ....	76
Stop Interrupt .....	471
Table of Interrupt Causes .....	605
Timer 00 Interrupt .....	237
Timer 01 Interrupt .....	237
Transfer Interrupt .....	470
Transmit interrupt.....	349, 401
Transmit Interrupt Generation and Flag Set Timing .....	406
Transmit Interrupt Request Generation Timing ...	407
Interrupt level setting register	
Interrupt Level Setting Registers (ILR0 to ILR5) Configuration.....	104
Interrupt Pin Selection Circuit	
Block Diagram of Interrupt Pin Selection Circuit	325
Interrupt Pin Selection Circuit .....	324
Operation of Interrupt Pin Selection Circuit .....	331
Pins Related to Interrupt Pin Selection Circuit.....	326
Registers Related to Interrupt Pin Selection Circuit .....	327
Interrupt pin selection circuit control register	
Interrupt pin selection circuit control register (WICR) .....	328
Interrupt request	
Interrupt Requests from Peripheral Resources .....	102
Interval	
Interrupt when Interval Function is in Operation .	158
Interval Timer	
Interrupt when Interval Timer Function is in Operation (Watch Interrupts) .....	182
Interval Timer Function .....	152, 176
Interval Timer Function (Continuous Mode).....	216
Interval Timer Function (Free-run Mode) .....	216
Interval Timer Function (One-shot Mode) .....	216

Operation of Interval Timer Function (Continuous Mode) .....	241
Operation of Interval Timer Function (Free-run Mode) .....	243
Operation of Interval Timer Function (One-shot Mode).....	239
Operation of Interval Timer Function (Watch prescaler) .....	184
When Interval Timer, Input Capture, or PWC Function Has Been Selected.....	254

## L

LCC-48P-M09	
Package Dimension of LCC-48P-M09 .....	16
LIN	
LIN Master Device .....	436
LIN Master/Slave Mode Communication Function .....	435
LIN Slave Device .....	437
LIN mode	
Asynchronous LIN Mode Operation .....	426
LIN synch field	
LIN Synch Field Edge Detection Interrupt (8/16-bit Compound Timer Interrupt)....	402
LIN-UART	
Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0).....	399
Bit Configuration of LIN-UART Extended Communication Control Register (ECCR) .....	397
Bit Configuration of LIN-UART Extended Status Control Register (ESCR) .....	395
Block Diagram of LIN-UART Pins .....	385
Functions of LIN-UART .....	378
LIN-UART Baud Rate Selection .....	408
LIN-UART Block Diagram .....	381
Operation of LIN-UART .....	416
LIN-UART Pin Direct Access.....	429
LIN-UART Reception Data Register (RDR).....	393
LIN-UART Reception Data Register (RDR/TDR) .....	393
LIN-UART serial control register (SCR) .....	387
LIN-UART serial status register (SSR) .....	391
LIN-UART Transmit Data Register (TDR) .....	394
Notes on Using LIN-UART .....	438
Pins related to LIN-UART .....	385
Register and Vector Table Related to LIN-UART Interrupt .....	403
Register List of LIN-UART .....	386
Sample Programs of LIN-UART .....	443
LIN-UART Baud Rate Generator Register 1, 0	
Bit Configuration of LIN-UART Baud Rate Generator Register 1, 0 (BGR1, BGR0).....	399

LIN-UART extended communication control register	
Bit Configuration of LIN-UART Extended	
Communication Control Register (ECCR)	397
LIN-UART extended status control register	
Bit Configuration of LIN-UART Extended Status	
Control Register (ESCR)	395
LIN-UART Reception Data Register	
LIN-UART Reception Data Register (RDR)	393
LIN-UART Reception Data Register	
(RDR/TDR)	393
LIN-UART serial mode register	
LIN-UART serial mode register (SMR)	389
LIN-UART serial status register	
LIN-UART serial status register (SSR)	391
LIN-UART Transmit Data Register	
LIN-UART Transmit Data Register (TDR)	394
List	
List of Registers of External Interrupt Circuit	314
Low-voltage Detection Reset Circuit	
Block Diagram of Low-voltage Detection Reset	
Circuit	511
Low-voltage Detection Reset Circuit	510
Operations of Low-voltage Detection Reset Circuit	513
Pins Related to Low-voltage Detection Reset Circuit	512
<b>M</b>	
Main clock	
Operation at the Main Clock Stop Mode	199
Main Clock Mode	
Operations in Main Clock Mode	69
Manipulation	
Read Destination on the Execution of Bit	
Manipulation Instructions	621
Mask Option List	
Mask Option List	626
Master	
LIN Master/Slave Mode Communication Function	435
Master/slave mode communication function	432
Master Device	
LIN Master Device	436
Master/Slave	
LIN Master/Slave Mode Communication Function	435
Master/slave mode communication function	432
MB2146-09	
Notes on F <sup>2</sup> MC-8FX Software Development Support	
Environment (MB95FV100D and MB2146-	
09)	562
MB95110B/M Series	
Block Diagram of All MB95110B/M Series	10
Feature of MB95110B/M Series	2
Pin Assignment of MB95110B/M Series	11
Product Lineup of MB95110B/M Series	4
MB95FV100D	
Notes on F <sup>2</sup> MC-8FX Software Development Support	
Environment (MB95FV100D and MB2146-	
09)	562
MB95FV100D and MB2146-09	
Notes on Software Programming Support Enviroment	
of F <sup>2</sup> MC-8FX (MB95FV100D and MB2146-	
09)	597
MCU	
Function to Wake Up the MCU	
from Standby Mode	481
MDSE	
One-shot Mode (MDSE of PCNTH0 Register:	
bit 5=1)	302
PWM Mode (MDSE of PCNTH Register:	
bit 5=0)	300
Memory	
Configuration of Memory Space	30
Memory Map	31, 33, 606
Placement of 16-bit Data in Memory	47
Sector Conversion Enable Bit in Flash Memory Status	
Register (FSR:SSEN)	593
Memory Map	
Memory Map	31, 33, 606
Memory Space	
Configuration of Memory Space	30
Mirror Address	
Register Bank Pointer, Direct Bank	
Pointer Mirror Address	40
Mode	
8/16-bit Compound Timer 00/01 Timer Mode Control	
Register ch.0 (TMCRO)	231
An Interrupt Request may Suppress Transition to	
Standby Mode.	76
Asynchronous LIN Mode Operation	426
Asynchronous Mode Operation	418
Check That Clock-mode Transition	
has been Completed before Setting Standby	
Mode.	76
Clock Mode State Transition Diagram	70
Clock Modes	53
Combinations of Clock Mode and Standby	
Mode	55
Function to Wake Up the MCU	
from Standby Mode	481
Interval Timer Function	
(Continuous Mode)	216
Interval Timer Function (Free-run Mode)	216
Interval Timer Function (One-shot Mode)	216
One-shot Mode (MDSE of PCNTH0 Register:	
bit 5=1)	302

Operating Description of UART/SIO Operation Mode 0 .....	351
Operating Description of UART/SIO Operation Mode 1 .....	358
Operation at the Main Clock Stop Mode .....	199
Operation in Sub Clock Stop Mode .....	199
Operation of 16-bit PPG Mode .....	280
Operation of 8-bit PPG Independent Mode .....	275
Operation of 8-bit Prescaler + 8-bit PPG Mode .....	277
Operation of Interval Timer Function (Continuous Mode) .....	241
Operation of Interval Timer Function (Free-run Mode) .....	243
Operation of Interval Timer Function (One-shot Mode) .....	239
Operation of PWM Timer Function (Fixed-cycle Mode) .....	245
Operation of PWM Timer Function (Variable-cycle Mode) .....	247
Operation of Synchronous Mode (Operation Mode 2) .....	422
Operations in Main Clock Mode .....	69
Operations in Main PLL Clock Mode .....	69
Operations in Sleep Mode .....	80
Operations in Stop Mode .....	81
Operations in Sub Clock Mode (on Dual Clock Product) .....	69
Operations in Sub PLL Clock Mode (on Dual Clock Product) .....	69
Operations in Time-base Timer Mode .....	82
Operations in Watch Mode .....	83
Oscillation Stabilization Wait Time and Clock Mode/Standby Mode Transition .....	57
Overview of Transitions to and from Standby Mode .....	75
Peripheral Resources not Affected by Clock Mode .....	53
Pin States in Standby Mode .....	75
Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction. ....	76
PWM Mode (MDSE of PCNTH Register: bit 5=0) .....	300
PWM Timer Function (Fixed-cycle Mode) .....	216
PWM Timer Function (Variable-cycle Mode) .....	216
Setting 16-bit PPG Mode .....	279
Setting 8-bit Independent Mode .....	275
Setting 8-bit Prescaler + 8-bit PPG Mode .....	277
Single-chip Mode .....	36
Standby Mode is Also Canceled when the CPU Rejects Interrupts. ....	76
Standby Mode State Transition Diagram .....	77
Standby Modes .....	54

## N

Nested Interrupts	
Nested Interrupts .....	108
Note	
Difference Points among Products and Notes on Selecting a Product .....	8
Note .....	557
Notes on Software Programming Support Environment of F <sup>2</sup> MC-8FX (MB95FV100D and MB2146-09) .....	597
Notes on Use of I <sup>2</sup> C .....	483
Notes	
Notes on Using 16-bit PPG Timer .....	304
Notes on Using 8/16-bit Compound Timer .....	256
Notes on Using 8/16-bit PPG .....	281
Notes on Using External Interrupt Circuit .....	320
O	
One-shot Mode	
Interval Timer Function (One-shot Mode) .....	216
One-shot Mode (MDSE of PCNTH0 Register: bit 5=1) .....	302
Operation of Interval Timer Function (One-shot Mode) .....	239
Operation	
Arithmetic Operation Instructions .....	623
Features of Dual Operation Flash .....	592
Interrupt During Operation of External Interrupt Circuit .....	317
Operation During Programming/Erasing .....	596
Operation of 16-bit PPG Mode .....	280
Operation of 8-bit PPG Independent Mode .....	275
Operation of 8-bit Prescaler + 8-bit PPG Mode .....	277
Operation of External Interrupt Circuit .....	318
Operation of I <sup>2</sup> C .....	473
Operation of Input Capture Function .....	251
Operation of Interval Timer Function (Continuous Mode) .....	241
Operation of Interval Timer Function (Free-run Mode) .....	243
Operation of Interval Timer Function (One-shot Mode) .....	239
Operation of PWC Timer Function .....	249
Operation of PWM Timer Function (Fixed-cycle Mode) .....	245
Operation of PWM Timer Function (Variable-cycle Mode) .....	247
Operations in Main Clock Mode .....	69
Operations in Main PLL Clock Mode .....	69
Operations in Sleep Mode .....	80
Operations in Stop Mode .....	81

Operations in Sub Clock Mode (on Dual Clock Product) .....	69
Operations in Sub PLL Clock Mode (on Dual Clock Product) .....	69
Operations in Time-base Timer Mode .....	82
Operations in Watch Mode .....	83
Operations of Port 0 .....	118
Operations of Port 1 .....	123
Operations of Port 2 .....	128
Operations of Port 3 .....	133
Operations of Port 5 .....	138
Operations of Port 6 .....	143
Operations of Port G .....	148
Operations of Prescaler.....	88
Read-modify-write Operation.....	621
Operation mode	
Operating Description of UART/SIO Operation Mode 0 .....	351
Operating Description of UART/SIO Operation Mode 1 .....	358
Operation of Synchronous Mode (Operation Mode 2) .....	422
Oscillation Stabilization Wait Time	
Configuration of Oscillation Stabilization Wait Time Setting Register (WATR).....	63
Oscillation Stabilization Wait Time .....	56
Oscillation Stabilization Wait Time and Clock Mode/ Standby Mode Transition.....	57
PLL Clock Oscillation Stabilization Wait Time .....	57
Oscillation Stabilization Wait Time Setting Register	
Configuration of Oscillation Stabilization Wait Time Setting Register (WATR).....	63
Other Instructions	
Other Instructions .....	624
Output	
8/16-bit PPG Output Inversion Register (REVC).....	272
Output Clock .....	87
Output Clock	
Output Clock .....	154, 178, 370
Overview	
Instruction Overview of F <sup>2</sup> MC-8FX .....	610
Overview of 8/16-bit PPG.....	258
Overview of Clock Controller .....	50
Overview of I/O Ports .....	114
Overview of Transitions to and from Standby Mode .....	75
<b>P</b>	
Package and Its Corresponding Product	
Package and Its Corresponding Product.....	9
Package Dimension	
Package Dimension of FPT-48P-M26 .....	15
Package Dimension of FPT-52P-M01 .....	14
Package Dimension of LCC-48P-M09 .....	16
Parallel Writer	
Writing to Flash Microcontroller Using Parallel Writer .....	628
PC	
8/16-bit PPG Timer 00 Control Register ch.0 (PC00).....	267
8/16-bit PPG Timer 01 Control Register ch.0 (PC01).....	265
PCNTH	
16-bit PPG Status Control Register, Upper (PCNTH0) .....	295
One-shot Mode (MDSE of PCNTH0 Register: bit 5=1).....	302
PWM Mode (MDSE of PCNTH Register: bit 5=0).....	300
PCNTL	
16-bit PPG Status Control Register, Lower (PCNTL0).....	297
PCSRH0, PCSRL0	
16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0).....	293
PDCRH0, PDCRL0	
16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0).....	292
PDS	
8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00) .....	270
PDUTH0, PDUTL0	
16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) .....	294
Peripheral function	
Interrupt Requests from Peripheral Resources .....	102
Peripheral Resources	
Peripheral Resources not Affected by Clock Mode.....	53
Pin	
Block Diagram of LIN-UART Pins .....	385
Block Diagram of Pins Related to 8/10-bit A/D Converter Block Diagram .....	494
Block Diagram of Pins Related to 8/16-bit Compound Timer .....	223
Block Diagram of Pins Related to 8/16-bit PPG.....	263
Block Diagram of Pins Related to External Interrupt Circuit .....	313
Block Diagram of Pins Related to UART/SIO .....	339
Block Diagrams of Pins Related to 16-bit PPG .....	290
Pin Connection .....	27
Pin State During a Reset.....	95
Pins of 16-bit PPG Timer .....	290
Pins of 8/10-bit A/D Converter .....	493



## INDEX

Pins of 8/16-bit PPG .....	262	Port 0 Register Function .....	117
Pins Related to 8/16-bit Compound Timer .....	222	Port 1	
Pins Related to External Interrupt Circuit.....	313	Block Diagram of Port 1 .....	121
Pins Related to I <sup>2</sup> C Bus Interface .....	455	Operations of Port 1 .....	123
Pins Related to Interrupt Pin Selection Circuit.....	326	Port 1 Configuration.....	120
Pins related to LIN-UART .....	385	Port 1 Pins.....	120
Pins Related to Low-voltage Detection Reset Circuit .....	512	Port 1 Register Function .....	122
Pins Related to UART/SIO.....	338	Port 2	
Port 0 Pins .....	115	Block Diagram of Port 2 .....	126
Port 1 Pins .....	120	Operations of Port 2 .....	128
Port 2 Pins .....	125	Port 2 Configuration.....	125
Port 3 Pins .....	130	Port 2 Pins.....	125
Port 5 Pins .....	135	Port 2 Register Function .....	127
Port 6 Pins .....	140	Port 3	
Port G Pins .....	145	Block Diagram of Port 3 .....	131
Pin Assignment		Operations of Port 3 .....	133
Pin Assignment of MB95110B/M Series .....	11	Port 3 Configuration.....	130
Pin Description		Port 3 Pins.....	130
Pin Description .....	17	Port 3 Register Function .....	132
LIN-UART Pin Direct Access		Port 5	
LIN-UART Pin Direct Access .....	429	Block Diagram of Port 5 .....	136
Pin State		Operations of Port 5 .....	138
Pin States in Standby Mode .....	75	Port 5 Configuration.....	135
Pin Status		Port 5 Pins.....	135
Pin Status in Each Mode .....	608	Port 5 Register Function .....	137
Place at Least Three NOP Instructions Immediately		Port 6	
Place at Least Three NOP Instructions Immediately		Block Diagram of Port 6.....	141
Following a Standby Mode Setting		Operations of Port 6 .....	143
Instruction. ....	76	Port 6 Configuration.....	140
Placement		Port 6 Pins.....	140
Placement of 16-bit Data in Memory .....	47	Port 6 Register Function .....	142
PLL		Port G	
Configuration of PLL Control Register		Block Diagram of Port G .....	146
(PLLCC).....	60	Operations of Port G .....	148
Operations in Main PLL Clock Mode.....	69	Port G Configuration.....	145
Operations in Sub PLL Clock Mode		Port G Pins .....	145
(on Dual Clock Product) .....	69	Port G Register Function .....	147
PLL Clock Oscillation Stabilization Wait		PPG	
Time .....	57	16-bit PPG Cycle Setting Buffer Registers	
PLL Clock		(Upper, Lower) (PCSRH0, PCSRL0)....	293
PLL Clock Oscillation Stabilization Wait		16-bit PPG Down Counter Registers	
Time .....	57	(Upper, Lower) (PDCRH0, PDCRL0)...	292
PLL Clock Mode		16-bit PPG Duty Setting Buffer Registers	
Operations in Main PLL Clock Mode.....	69	(Upper, Lower) (PDUTH0, PDUTL0)...	294
Operations in Sub PLL Clock Mode		16-bit PPG Status Control Register, Lower	
(on Dual Clock Product) .....	69	(PCNTL0).....	297
PLL Control Register		16-bit PPG Status Control Register, Upper	
Configuration of PLL Control Register		(PCNTH0) .....	295
(PLLCC).....	60	16-bit PPG Timer.....	286
Port 0		8/16-bit PPG Output Inversion Register	
Block Diagram of Port 0 .....	116	(REVC) .....	272
Operations of Port 0.....	118	8/16-bit PPG Start Register (PPGS) .....	271
Port 0 Configuration .....	115	8/16-bit PPG Timer 00 Control Register ch.0	
Port 0 Pins .....	115	(PC00) .....	267
		8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register	
		(PPS01), (PPS00).....	269

8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00).....	270
8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
Block Diagram of 16-bit PPG Timer .....	287
Block Diagram of 8/16-bit PPG .....	259
Block Diagram of Pins Related to 8/16-bit PPG .....	263
Block Diagrams of Pins Related to 16-bit PPG .....	290
Channels of 16-bit PPG Timer .....	289
Channels of 8/16-bit PPG .....	261
Interrupts of 16-bit PPG Timer .....	299
Interrupts of 8/16-bit PPG .....	273
Notes on Using 16-bit PPG Timer .....	304
Notes on Using 8/16-bit PPG .....	281
Operation of 16-bit PPG Mode .....	280
Operation of 8-bit PPG Independent Mode .....	275
Operation of 8-bit Prescaler + 8-bit PPG Mode .....	277
Overview of 8/16-bit PPG .....	258
Pins of 16-bit PPG Timer .....	290
Pins of 8/16-bit PPG .....	262
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers and Vector Table Related to Interrupts of 8/16-bit PPG .....	273
Registers of 16-bit PPG Timer .....	291
Registers of 8/16-bit PPG .....	264
Sample Programs for 16-bit PPG Timer .....	305
Sample Programs for 8/16-bit PPG Timer .....	282
Setting 16-bit PPG Mode .....	279
Setting 8-bit Prescaler + 8-bit PPG Mode .....	277
<b>PPG Timer</b>	
16-bit PPG Timer .....	286
8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00).....	269
8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00).....	270
8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
Block Diagram of 16-bit PPG Timer .....	287
Channels of 16-bit PPG Timer .....	289
Interrupts of 16-bit PPG Timer .....	299
Notes on Using 16-bit PPG Timer .....	304
Pins of 16-bit PPG Timer .....	290
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers of 16-bit PPG Timer .....	291
Sample Programs for 16-bit PPG Timer .....	305
Sample Programs for 8/16-bit PPG Timer .....	282
<b>PPGS</b>	
8/16-bit PPG Start Register (PPGS) .....	271
<b>PPS</b>	
8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00) .....	269
<b>Prescaler</b>	
Block Diagram of Watch Prescaler .....	177
Operation of 8-bit Prescaler + 8-bit PPG Mode .....	277
Operations of Prescaler .....	88
Prescaler .....	86
Prescaler Block Diagram .....	87
Setting 8-bit Prescaler + 8-bit PPG Mode .....	277
<b>Procedure</b>	
Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN).....	595
Setup Procedure Example .....	274, 303, 319, 473
<b>Product Lineup</b>	
Product Lineup of MB95110B/M Series .....	4
<b>Program</b>	
I <sup>2</sup> C Sample Programs .....	485
Sample Programs for 16-bit PPG Timer .....	305
Sample Programs for 8/16-bit PPG Timer .....	282
Sample Programs for External Interrupt Circuit .....	321
Setting Methods not Covered by Sample Programs .....	187, 201, 364, 443, 506
Setting Methods Other than Those in Sample Programs .....	485
Setup Methods without Sample Program .....	282, 305, 321
<b>Programming</b>	
Interrupt occurrence when programming upper bank .....	559
Notes on Software Programming Support Environment of F <sup>2</sup> MC-8FX (MB95FV100D and MB2146-09) .....	597
Operation During Programming/Erasing .....	596
<b>Programming Procedure</b>	
Flash Memory Programming Procedure .....	552, 576
<b>PSSR</b>	
UART/SIO Dedicated Baud Rate Generator Prescaler Selection Register (PSSR0) .....	373
<b>PWC</b>	
Operation of PWC Timer Function .....	249
PWC Timer Function .....	217
When Interval Timer, Input Capture, or PWC Function Has Been Selected .....	254
<b>PWM</b>	
Operation of PWM Timer Function (Fixed-cycle Mode).....	245
Operation of PWM Timer Function (Variable-cycle Mode) .....	247
PWM Mode (MDSE of PCNTH Register: bit 5=0).....	300
PWM Timer Function (Fixed-cycle Mode).....	216

## INDEX

PWM Timer Function (Variable-cycle Mode) .....	216
PWM Timer	
Operation of PWM Timer Function (Fixed-cycle Mode) .....	245
Operation of PWM Timer Function (Variable-cycle Mode) .....	247
PWM Timer Function (Fixed-cycle Mode) .....	216
PWM Timer Function (Variable-cycle Mode) .....	216
<b>R</b>	
RAM	
Effect of Reset on RAM Contents .....	94
RDR	
LIN-UART Reception Data Register (RDR) .....	393
LIN-UART Reception Data Register (RDR/TDR) .....	393
UART/SIO Serial Input Data Register (RDR0) .....	347
Read	
Placing Flash Memory in the Read/Reset State .....	551, 575
Read Destination on the Execution of Bit Manipulation Instructions.....	621
Read-modify-write	
Read-modify-write Operation .....	621
Reception	
Reception interrupt .....	349, 400
Reception Interrupt Generation and Flag Set Timing .....	404
Register	
16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0) ....	293
16-bit PPG Down Counter Registers (Upper, Lower) (PDCRH0, PDCRL0) ...	292
16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) ...	294
16-bit PPG Status Control Register, Lower (PCNTL0) .....	297
16-bit PPG Status Control Register, Upper (PCNTH0) .....	295
8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) .....	225
8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) .....	228
8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR) .....	234
8/16-bit Compound Timer 00/01 Timer Mode Control Register ch.0 (TMCRO).....	231
8/16-bit PPG Output Inversion Register (REVC) .....	272
8/16-bit PPG Start Register (PPGS) .....	271
8/16-bit PPG Timer 00 Control Register ch.0 (PC00) .....	267
8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00) .....	269
8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00) .....	270
8/16-bit PPG Timer 01 Control Register ch.0 (PC01) .....	265
Configuration of Oscillation Stabilization Wait Time Setting Register (WATR).....	63
Configuration of PLL Control Register (PLLC) .....	60
Configuration of System Clock Control Register (SYCC).....	58
External Interrupt Control Register (EIC00) .....	315
General-purpose Register Area (Addresses: 0100 <sub>H</sub> to 01FF <sub>H</sub> ) .....	32
I <sup>2</sup> C Address Register (IAAR0) .....	467
I <sup>2</sup> C Bus Control Register 0 (IBCR00) .....	458
I <sup>2</sup> C Bus Control Register 1 (IBCR10) .....	461
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
I <sup>2</sup> C Clock Control Register (ICCR0).....	468
I <sup>2</sup> C Data Register (IDDR0) .....	466
I <sup>2</sup> C Registers .....	457
List of Registers of External Interrupt Circuit .....	314
One-shot Mode (MDSE of PCNTH0 Register: bit 5=1) .....	302
Port 0 Register Function .....	117
Port 1 Register Function .....	122
Port 2 Register Function .....	127
Port 3 Register Function .....	132
Port 5 Register Function .....	137
Port 6 Register Function .....	142
Port G Register Function .....	147
PWM Mode (MDSE of PCNTH Register: bit 5=0) .....	300
Registers and Vector Table Related to I <sup>2</sup> C Interrupts .....	472
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers and Vector Table Related to Interrupts of 8/16-bit PPG .....	273
Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer.....	238
Registers of 16-bit PPG Timer .....	291
Registers of 8/16-bit PPG .....	264
Registers Related to 8/16-bit Compound Timer... ..	224
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSSEN) .....	593
Standby Control Register (STBC).....	66
Register Bank Pointer	
Configuration of Register Bank Pointer (RP) .....	40
Register Bank Pointer, Direct Bank Pointer Mirror Address .....	40
Reload Counter	
Function of Reload Counter .....	414

Operation of Dedicated Baud Rate Generator (Reload Counter).....	413
Reload value	
Reload Value and Baud Rate of Each Clock Speed .....	411
Reset	
Block Diagram of Low-voltage Detection Reset Circuit.....	511
Effect of Reset on RAM Contents.....	94
Low-voltage Detection Reset Circuit .....	510
Notes on Using Reset .....	99
Overview of Reset Operation .....	94
Pin State During a Reset .....	95
Placing Flash Memory in the Read/Reset State .....	551, 575
Reset output .....	93
Reset Sources .....	92
Reset Time .....	93
Reset cause register	
Configuration of Reset Source Register (RSRR) ...	96
Status of Reset Source Register (RSRR).....	98
Resuming	
Resuming Sector Erasing from Flash Memory ....	558
REVC	
8/16-bit PPG Output Inversion Register (REVC) .....	272
RP	
Configuration of Register Bank Pointer (RP) .....	40
RSRR	
Configuration of Reset Source Register (RSRR) ...	96
Status of Reset Source Register (RSRR).....	98
<b>S</b>	
Sample	
I <sup>2</sup> C Sample Programs .....	485
Sample Programs for 16-bit PPG Timer .....	305
Sample Programs for 8/16-bit PPG Timer .....	282
Sample Programs for External Interrupt Circuit.....	321
Setting Methods Other than Those in Sample Programs.....	485
Setup Methods without Sample Program .....	282, 305, 321
Sample Program	
Setup Methods without Sample Program .....	282, 305, 321
Sample Programs	
Sample Programs for 8/10-bit A/D Converter .....	506
Sample Programs for UART/SIO .....	364
Sample Programs for Watch Counter .....	201
Sample Programs for Watch Prescaler.....	187
Sample Programs of LIN-UART .....	443
SCR	
LIN-UART serial control register (SCR) .....	387
Sector	
Sector Configuration of 256-Kbit Flash Memory .....	565
Sector/Bank Configuration of Dual-Operation Flash Memory .....	530
Sector Configuration	
Sector Configuration.....	629
Sector Conversion Enable Bit	
Access Sector Map Based on Sector Conversion Enable Bit (FSR:SSEN) .....	594
Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN).....	595
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN).....	593
Setup Procedure of Sector Conversion Enable Bit (FSR:SSEN) Setup Procedure .....	559
Sector Erase Timer Flag	
Sector Erase Timer Flag (DQ3).....	549
Sector erasing	
Erasing Arbitrary Data from Flash Memory (Sector Erasing).....	555
Flash Memory Sector Erasing Procedure.....	555
Notes on Erasing Data from Sectors .....	555
Resuming Sector Erasing from Flash Memory.....	558
Suspending Sector Erasing from Flash Memory.....	557
Selecting	
Difference Points among Products and Notes on Selecting a Product .....	8
Sequence	
Command Sequence Table .....	540, 569
Serial Programming	
Basic Configuration of Serial Programming Connection for Flash Memory Products .....	582
Example of Serial Programming Connection .....	585
Setting	
16-bit PPG Cycle Setting Buffer Registers (Upper, Lower) (PCSRH0, PCSRL0).....	293
16-bit PPG Duty Setting Buffer Registers (Upper, Lower) (PDUTH0, PDUTL0) ...	294
Check That Clock-mode Transition has been Completed before Setting Standby Mode.....	76
Configuration of Oscillation Stabilization Wait Time Setting Register (WATR) .....	63
Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction.....	76
Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN).....	595
Setting 16-bit PPG Mode.....	279
Setting 8-bit Independent Mode .....	275
Setting 8-bit Prescaler + 8-bit PPG Mode.....	277
Setting Methods Other than Those in Sample Programs .....	485

## INDEX

Setup	
8/16-bit PPG Timer 00/01 Cycle Setup Buffer Register (PPS01), (PPS00) .....	269
8/16-bit PPG Timer 00/01 Duty Setup Buffer Register (PDS01), (PDS00) .....	270
Setup Methods without Sample Program .....	282, 305, 321
Setup Procedure Example .....	274, 303, 319, 473
Setup Flowchart	
Flash Memory Sector Write Control Register (SWRE0/SWRE1) Setup Flowchart .....	538
Setup Procedure Example	
Setup Procedure Example .....	162, 173, 185, 199, 350, 417, 504
Signaling	
Signaling .....	417
Single-chip Mode	
Single-chip Mode .....	36
Slave	
LIN Master/Slave Mode Communication Function .....	435
Master/slave mode communication function .....	432
Slave Device	
LIN Slave Device .....	437
Sleep	
Operations in Sleep Mode .....	80
Sleep Mode	
Operations in Sleep Mode .....	80
SMC	
UART/SIO Serial Mode Control Register 1 (SMC10) .....	341
UART/SIO Serial Mode Control Register 2 (SMC20) .....	343
SMR	
LIN-UART serial mode register (SMR) .....	389
Software	
Notes on Software Programming Support Environment of F <sup>2</sup> MC-8FX (MB95FV100D and MB2146-09) .....	597
Special Instruction	
Special Instruction .....	617
SSEN	
Access Sector Map Based on Sector Conversion Enable Bit (FSR:SSEN) .....	594
Procedure of Setting the Sector Conversion Enable Bit (FSR:SSEN) .....	595
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SSEN) .....	593
Setup Procedure of Sector Conversion Enable Bit (FSR:SSEN) .....	559
SSR	
LIN-UART serial status register (SSR) .....	391
UART/SIO serial status and data register (SSR0) .....	345
Stack	
Interrupt Processing Stack Area .....	111
Stack Operation at Start of Interrupt Processing ..	110
Stack Operation upon Returning from Interrupt ..	110
Standby	
An Interrupt Request may Suppress Transition to Standby Mode. ....	76
Check That Clock-mode Transition has been Completed before Setting Standby Mode. ....	76
Combinations of Clock Mode and Standby Mode .....	55
Function to Wake Up the MCU from Standby Mode .....	481
Oscillation Stabilization Wait Time and Clock Mode/ Standby Mode Transition .....	57
Overview of Transitions to and from Standby Mode .....	75
Pin States in Standby Mode .....	75
Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction. ....	76
Standby Control Register (STBC) .....	66
Standby Mode is Also Canceled when the CPU Rejects Interrupts. ....	76
Standby Mode State Transition Diagram .....	77
Standby Modes .....	54
Standby Control Register	
Standby Control Register (STBC) .....	66
Standby Mode	
An Interrupt Request may Suppress Transition to Standby Mode. ....	76
Check That Clock-mode Transition has been Completed before Setting Standby Mode. ....	76
Combinations of Clock Mode and Standby Mode .....	55
Function to Wake Up the MCU from Standby Mode .....	481
Operations in Standby Mode .....	513
Oscillation Stabilization Wait Time and Clock Mode/ Standby Mode Transition .....	57
Overview of Transitions to and from Standby Mode .....	75
Pin States in Standby Mode .....	75
Place at Least Three NOP Instructions Immediately Following a Standby Mode Setting Instruction. ....	76
Standby Mode is Also Canceled when the CPU Rejects Interrupts. ....	76
Standby Mode State Transition Diagram .....	77
Standby Modes .....	54
Start	
8/16-bit PPG Start Register (PPGS) .....	271

Start Conditions .....	475
Start Conditions	
Start Conditions .....	475
Startup	
Example Startup Flowchart when using the Clock Supervisor .....	524
State	
Clock Mode State Transition Diagram .....	70
Standby Mode State Transition Diagram .....	77
Status	
16-bit PPG Status Control Register, Lower (PCNTLO) .....	297
16-bit PPG Status Control Register, Upper (PCNTH0) .....	295
8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) .....	225
8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) .....	228
I <sup>2</sup> C Bus Status Register (IBSR0) .....	464
Sector Conversion Enable Bit in Flash Memory Status Register (FSR:SEN) .....	593
Status Register	
8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) .....	228
STBC	
Standby Control Register (STBC) .....	66
Stop	
Operations in Stop Mode .....	81
Stop Condition .....	478
Stop Interrupt .....	471
Stop Mode	
Operations in Stop Mode .....	81
Stop mode	
Operation at the Main Clock Stop Mode .....	199
Operation in Sub Clock Stop Mode .....	199
Sub Clock	
Operation in Sub Clock Stop Mode .....	199
Sub Clock Mode	
Operations in Sub Clock Mode (on Dual Clock Product) .....	69
Suspending	
Suspending Sector Erasing from Flash Memory ..	557
SWRE	
Flash Memory Sector Write Control Registers (SWRE0/SWRE1) .....	535
Flash Memory Sector Write Control Register (SWRE0/SWRE1) Setup Flowchart .....	538
SYCC	
Configuration of System Clock Control Register (SYCC) .....	58
Synchronous method	
Synchronous method .....	417
Synchronous Mode	
Operation of Synchronous Mode (Operation Mode 2) .....	422
System	
Configuration of System Clock Control Register (SYCC) .....	58
I <sup>2</sup> C System .....	474
System Clock Control Register	
Configuration of System Clock Control Register (SYCC) .....	58
T	
T00CR0/T01CR0	
8/16-bit Compound Timer 00/01 Control Status Register 0 (T00CR0/T01CR0) .....	225
T00CR1/T01CR1	
8/16-bit Compound Timer 00/01 Control Status Register 1 (T00CR1/T01CR1) .....	228
T00DR/T01DR	
8/16-bit Compound Timer 00/01 Data Register (T00DR/T01DR) .....	234
Table	
Explanation of Item in Instruction Table .....	612
Registers and Vector Table Related to I <sup>2</sup> C Interrupts .....	472
Registers and Vector Table Related to Interrupts of 16-bit PPG Timer .....	299
Registers and Vector Table Related to Interrupts of 8/16-bit PPG .....	273
Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Table of Interrupt Causes .....	605
Vector Table Area (Addresses: FFC0 <sub>H</sub> to FFFF <sub>H</sub> ) .....	32
TBTC	
Time-base Timer Control Register (TBTC) .....	156
TDR	
LIN-UART Reception Data Register (RDR/TDR) .....	393
LIN-UART Transmit Data Register (TDR) .....	394
UART/SIO Serial Output Data Register (TDR0) .....	348
Time counter data register	
Watch Counter Data Register (WCDR) .....	194
Time-base timer	
Block Diagram of Time-base Timer .....	153
Clearing Time-base Timer .....	161
Notes on Using Time-base Timer .....	163
Operating Examples of Time-base Timer .....	161
Operations of Time-base Timer .....	160
Register and Vector Table for Interrupts of Time-base Timer .....	159
Registers of the Time-base Timer .....	155

## INDEX

Time-base timer control register	
Time-base Timer Control Register (TBTC).....	156
Time-base Timer Mode	
Operations in Time-base Timer Mode .....	82
Timer 00	
Timer 00 Interrupt .....	237
Timer 01	
Timer 01 Interrupt .....	237
Timing	
Reception Interrupt Generation and Flag Set Timing	
.....	404
Transmit Interrupt Generation and Flag Set Timing	
.....	406
Transmit Interrupt Request Generation Timing ...	407
Timing Limit Elapsed Flag	
Timing Limit Elapsed Flag	
(DQ5) .....	548, 573
TMCR	
8/16-bit Compound Timer 00/01 Timer Mode Control	
Register ch.0 (TMCR0).....	231
Toggle Bit Flag	
Restrictions on Toggle Bit Flag (DQ6) .....	562
Toggle Bit Flag (DQ6).....	546, 572
Transfer	
Data Transfer .....	475
Transfer Instructions.....	622
Transfer Interrupt .....	470
Transfer Instructions	
Transfer Instructions.....	622
Transition	
An Interrupt Request may Suppress Transition to	
Standby Mode. ....	76
Check That Clock-mode Transition	
has been Completed before Setting Standby	
Mode.....	76
Clock Mode State Transition Diagram.....	70
Oscillation Stabilization Wait Time and Clock Mode/	
Standby Mode Transition .....	57
Overview of Transitions to and from Standby	
Mode .....	75
Standby Mode State Transition Diagram .....	77
Transmission	
Transmit interrupt.....	349, 401
Transmit Interrupt Generation and Flag Set Timing	
.....	406
Transmit Interrupt Request Generation Timing ...	407

## U

UART/SIO	
Block Diagram of Pins Related to UART/SIO.....	339
Block Diagram of UART/SIO .....	335
Channels of UART/SIO .....	337
Functions of UART/SIO .....	334
Interrupts of UART/SIO .....	349

Operating Description of UART/SIO Operation Mode	
0 .....	351
Operating Description of UART/SIO Operation Mode	
1 .....	358
Operation of UART/SIO.....	350
Pins Related to UART/SIO .....	338
Registers and Vector Table Related to UART/SIO	
Interrupts .....	349
Registers Related to UART/SIO .....	340
Sample Programs for UART/SIO .....	364

UART/SIO Dedicated Baud Rate Generator	
Block Diagram of UART/SIO Dedicated Baud Rate	
Generator .....	370
Channels of UART/SIO Dedicated Baud Rate	
Generator .....	371
Registers Related to UART/SIO Dedicated Baud Rate	
Generator .....	372

UART/SIO Serial Input Data Register	
UART/SIO Serial Input Data Register	
(RDR0).....	347

UART/SIO Serial Mode Control Register	
UART/SIO Serial Mode Control Register 1 (SMC10)	
.....	341
UART/SIO Serial Mode Control Register 2 (SMC20)	
.....	343

UART/SIO Serial Output Data Register	
UART/SIO Serial Output Data Register	
(TDR0) .....	348

UART/SIO serial status and data Registers	
UART/SIO serial status and data register (SSR0)	345

## V

Variable-cycle Mode	
Operation of PWM Timer Function	
(Variable-cycle Mode).....	247
PWM Timer Function	
(Variable-cycle Mode).....	216

Vector Table	
Register and Vector Table for Interrupts of Time-base	
Timer.....	159
Register and Vector Table Related to 8/10-bit A/D	
Converter Interrupts .....	501
Register and Vector Table Related to Interrupts of	
Watch Counter.....	197
Register and Vector Table Related to Interrupts of	
Watch Prescaler .....	183
Register and Vector Table Related to LIN-UART	
Interrupt.....	403
Registers and Vector Table Related to I <sup>2</sup> C	
Interrupts .....	472
Registers and Vector Table Related to Interrupts	
of 16-bit PPG Timer .....	299
Registers and Vector Table Related to Interrupts	
of 8/16-bit PPG.....	273

Registers and Vector Table Related to Interrupts of External Interrupt Circuit .....	317
Registers and Vector Table Related to UART/SIO Interrupts.....	349
Registers and Vector Tables Related to Interrupts of 8/16-bit Compound Timer.....	238
Vector Table Area (Addresses: FFC0 <sub>H</sub> to FFFF <sub>H</sub> ).....	32
<b>W</b>	
Watch counter	
Block Diagram of Watch Counter .....	191
Interrupts of Watch Counter .....	197
Register and Vector Table Related to Interrupts of Watch Counter .....	197
Registers of Watch Counter .....	193
Sample Programs for Watch Counter .....	201
Setup Procedure of Watch Counter .....	198
Watch counter .....	190
Watch counter control register	
Watch Counter Control Register (WCSR) .....	195
Watch Interrupts	
Interrupt when Interval Timer Function is in Operation (Watch Interrupts) .....	182
Watch Mode	
Operations in Watch Mode .....	83
Watch prescaler	
Block Diagram of Watch Prescaler .....	177
Clearing Watch Prescaler .....	184
Interrupts of Watch Prescaler .....	182
Notes on Using Watch Prescaler.....	186
Operating Examples of Watch Prescaler.....	184
Operation of Interval Timer Function (Watch prescaler) .....	184
Register and Vector Table Related to Interrupts of Watch Prescaler .....	183
Register of the Watch Prescaler .....	179
Sample Programs for Watch Prescaler.....	187
Watch prescaler control register	
Watch Prescaler Control Register (WPCR).....	180
Watchdog timer	
Block Diagram of Watchdog Timer .....	167
Notes on Using Watchdog Timer .....	174
Operations of Watchdog Timer .....	172
Register of The Watchdog Timer .....	169
Watchdog Timer Function .....	166
Watchdog timer control register	
Watchdog timer control register (WDTC).....	170
WCDR	
Watch Counter Data Register (WCDR) .....	194
WCSR	
Watch Counter Control Register (WCSR) .....	195
WDTC	
Watchdog timer control register (WDTC) .....	170
WICR	
Interrupt pin selection circuit control register (WICR) .....	328
Wild register address compare enable register	
Wild Register Address Compare Enable Register (WREN) .....	211
Wild Register Address Setup Registers	
Wild Register Address Setup Registers (WRAR0 to WRAR2) .....	210
Wild Register Data Setup Registers	
Wild Register Data Setup Registers (WRDR0 to WRDR2) .....	209
Wild register data test setting register	
Wild Register Data Test Setup Register (WROR) .....	212
Wild Registers	
Block Diagram of Wild Register Function .....	205
Registers Related to Wild Register .....	207
Setup Procedure for Wild Register .....	213
Wild Register Applicable Addresses.....	213
Wild Register Function .....	204
Wild register number .....	208
WPCR	
Watch Prescaler Control Register (WPCR) .....	180
WRAR	
Wild Register Address Setup Registers (WRAR0 to WRAR2) .....	210
WRDR	
Wild Register Data Setup Registers (WRDR0 to WRDR2) .....	209
WRE	
Note on Setting the FSR:WRE .....	539
WREN	
Wild Register Address Compare Enable Register (WREN) .....	211
Write	
Details of Programming/Erasing Flash Memory .....	550, 574
Flash memory program/erase .....	564
Flash Memory Programming/Erasing .....	529
How to Write .....	629
Programming Data into Flash Memory .....	552
Programming Data into Flash Memory Write .....	576
Programming/Erasing Operation .....	560
Writing	
Writing to Flash Microcontroller Using Parallel Writer .....	628
WROR	
Wild Register Data Test Setup Register (WROR) .....	212





# Register Index

## A

AIDRL A/D input disable register lower ..... 114

## D

DDR0 Port 0 direction register..... 114  
 DDR1 Port 1 direction register..... 114  
 DDR2 Port 2 direction register..... 114  
 DDR3 Port 3 direction register..... 114  
 DDR5 Port 5 direction register..... 114  
 DDR6 Port 6 direction register..... 114  
 DDRG Port G direction register..... 114

## E

EIC00 External interrupt control register ch.0/ch.1  
 ..... 315  
 EIC10 External interrupt control register ch.2/ch.3  
 ..... 315  
 EIC20 External interrupt control register ch.4/ch.5  
 ..... 315  
 EIC30 External interrupt control register ch.6/ch.7  
 ..... 315

## I

IAAR0 I<sup>2</sup>C address register ch.0 ..... 467  
 IAAR1 I<sup>2</sup>C address register ch.1 ..... 467  
 IBCR00 I<sup>2</sup>C bus control register 0 ch.0 ..... 458  
 IBCR10 I<sup>2</sup>C bus control register 1 ch.0 ..... 458  
 IBSR0 I<sup>2</sup>C bus status register ch.0 ..... 464  
 ICCR0 I<sup>2</sup>C clock control register ch.0..... 468  
 IDDR0 I<sup>2</sup>C data register ch.0..... 466  
 ILSR Input level selection register ..... 114  
 ILSR2 Input level selection register 2 ..... 114

## P

PC00 8/16-bit PPG timer 00 control register ch.0  
 ..... 267  
 PC01 8/16-bit PPG timer 01 control register ch.0  
 ..... 265  
 PC10 8/16-bit PPG timer 00 control register ch.1  
 ..... 267  
 PC11 8/16-bit PPG timer 01 control register ch.1  
 ..... 265  
 PCNTH0 16-bit PPG status control register upper  
 ch.0 ..... 295

PCNTL0 16-bit PPG status control register lower  
 ch.0.....295  
 PCSRH0 16-bit PPG cycle setting buffer register  
 upper ch.0 .....293  
 PCSRL0 16-bit PPG cycle setting buffer register  
 lower ch.0 .....293  
 PDCRH0 16-bit PPG down counter register upper  
 ch.0.....292  
 PDCRL0 16-bit PPG down counter register lower  
 ch.0.....292  
 PDR0 Port 0 data register ..... 114  
 PDR1 Port 1 data register ..... 114  
 PDR2 Port 2 data register ..... 114  
 PDR3 Port 3 data register ..... 114  
 PDR5 Port 5 data register ..... 114  
 PDR6 Port 6 data register ..... 114  
 PDRG Port G data register ..... 114  
 PDS00 8/16-bit PPG timer 00 duty setting buffer  
 register ch.0.....270  
 PDS01 8/16-bit PPG timer 01 duty setting buffer  
 register ch.0.....270  
 PDS10 8/16-bit PPG timer 00 duty setting buffer  
 register ch.1.....270  
 PDS11 8/16-bit PPG timer 01 duty setting buffer  
 register ch.1.....270  
 PDUTH0 16-bit PPG duty setting buffer register  
 upper ch.0 .....294  
 PDUTL0 16-bit PPG duty setting buffer register lower  
 ch.0.....294  
 PLLC PLL control register .....60  
 PPGS 8/16-bit PPG start register .....271  
 PPS00 8/16-bit PPG timer 00 cycle setup buffer  
 register ch.0.....269  
 PPS01 8/16-bit PPG timer 01 cycle setup buffer  
 register ch.0.....269  
 PPS10 8/16-bit PPG timer 00 cycle setup buffer  
 register ch.1.....269  
 PPS11 8/16-bit PPG timer 01 cycle setup buffer  
 register ch.1.....269  
 PUL1 Port 1 pull-up control register ..... 114  
 PUL2 Port 2 pull-up control register ..... 114  
 PUL3 Port 3 pull-up control register ..... 114  
 PULG Port G pull-up control register ..... 114

## Register Index

### R

REVC 8/16-bit PPG output inversion register .....272

### S

STBC Standby control register .....66

SYCC System clock control register .....58

### T

T00CR0 8/16-bit compond timer 00 control status  
register 0 ch.0 .....225

T00CR1 8/16-bit compond timer 00 control status  
register 1 ch.0 .....228

T00DR 8/16-bit compond timer 00 data register  
ch.0 .....234

T01CR0 8/16-bit compond timer 01 control status  
register 0 ch.0 .....225

T01CR1 8/16-bit compond timer 01 control status  
register 1 ch.0 .....228

T01DR 8/16-bit compond timer 01 data register  
ch.0 .....234

T10CR0 8/16-bit compond timer 10 control status  
register 0 ch.1 .....225

T10CR1 8/16-bit compond timer 10 control status  
register 1 ch.1 .....228

T10DR 8/16-bit compond timer 10 data register  
ch.1 .....234

T11CR0 8/16-bit compond timer 11 control status  
register 0 ch.1 .....225

T11CR1 8/16-bit compond timer 11 control status  
register 1 ch.1 .....228

T11DR 8/16-bit compond timer 11 data register  
ch.1 .....234

TMCR0 8/16-bit compond timer 00/01 timer mode  
control register ch.0 .....231

TMCR1 8/16-bit compond timer 00/01 timer mode  
control register ch.1 .....231

### W

WATR Oscillation stabilization wait time setting  
register .....63





# Interrupt Vector Index

## I

IRQ0	External interrupt ch.0.....	317
IRQ0	External interrupt ch.4.....	317
IRQ1	External interrupt ch.1.....	317
IRQ1	External interrupt ch.5.....	317
IRQ10	8/16-bit PPG ch.1 upper .....	273
IRQ12	8/16-bit PPG ch.0 upper .....	273
IRQ13	8/16-bit PPG ch.0 lower.....	273
IRQ14	8/16-bit compond timer ch.1 upper.....	238
IRQ15	16-bit PPG ch.0 .....	299
IRQ16	I <sup>2</sup> C ch.0.....	472
IRQ2	External interrupt ch.2.....	317
IRQ2	External interrupt ch.6.....	317
IRQ22	8/16-bit compond timer ch.1 lower.....	238
IRQ23	Flash memory .....	605
IRQ3	External interrupt ch.3.....	317
IRQ3	External interrupt ch.7.....	317
IRQ5	8/16-bit compond timer ch.0 lower.....	238
IRQ6	8/16-bit compond timer ch.0 upper.....	238
IRQ9	8/16-bit PPG ch.1 lower.....	273



CM26-10109-4E

---

**FUJITSU MICROELECTRONICS • CONTROLLER MANUAL**

F<sup>2</sup>MC-8FX

8-BIT MICROCONTROLLER

MB95110B/M Series

HARDWARE MANUAL

---

February 2010 the fourth edition

Published **FUJITSU MICROELECTRONICS LIMITED**

Edited Sales Promotion Dept.

---



