



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.

F²MCTM-8L
8-BIT MICROCONTROLLER
MB89580B/BW Series
HARDWARE MANUAL

F²MCTM-8L
8-BIT MICROCONTROLLER
MB89580B/BW Series
HARDWARE MANUAL

FUJITSU LIMITED

PREFACE

■ Objectives and Intended Reader

The MB89580B/BW series of products has been developed as one of the general products of F²MC-8L family. The F²MC-8L family is an original 8 bit one-chip microcontroller that can be used for Application Specific ICs (ASICs). The MB89580B/BW series has a wide range of uses in consumer and industrial equipment.

This manual covers the functions and operations of the MB89580B/BW series and is designed for engineers who develop products using the MB89580B/BW series of microcontrollers. For information on the microcontroller's instruction set, refer to the F²MC-8L Programming Manual.

■ Trademarks

F²MC is the abbreviation of FUJITSU Flexible Microcontroller.

Other system and product names in this manual are trademarks of respective companies or organizations.

The symbols TM and ® are sometimes omitted in this manual.

■ Structure of This Manual

This manual consists of the following 13 chapters and appendix.

CHAPTER 1 "OVERVIEW"

This chapter explains the features and basic specifications of the MB89580B/BW series of microcontrollers.

CHAPTER 2 "HANDLING DEVICE"

This chapter explains the precautions to be taken when using the USB general-purpose one-chip microcontroller.

CHAPTER 3 "CENTRAL PROCESSING UNIT (CPU)"

This chapter explains the functions and operation of the CPU.

CHAPTER 4 "I/O PORTS"

This chapter explains the functions and operation of the I/O ports.

CHAPTER 5 "TIMEBASE TIMER"

This chapter explains the functions and operation of the timebase timer.

CHAPTER 6 "WATCHDOG TIMER"

This chapter explains the functions and operation of the watchdog timer.

CHAPTER 7 "2-CHANNEL 8-BIT PWM TIMER"

This chapter explains the functions and operation of the 2-channel 8-bit PWM timer.

CHAPTER 8 "EXTERNAL INTERRUPT CIRCUIT (LEVEL)"

This chapter explains the functions and operation of the external interrupt circuit (level).

CHAPTER 9 "PARALLEL PORTS"

This chapter explains the functions and operation of the parallel ports.

CHAPTER 10 "USB FUNCTION"

This chapter explains the functions and operation of the USB function circuit.

CHAPTER 11 "UART/SIO"

This chapter explains the functions and operation of UART/SIO.

CHAPTER 12 "CLOCK OUTPUT FUNCTION"

This chapter explains the clock output function and the circuit operation.

CHAPTER 13 "PULL-UP OPTION"

This chapter explains the pull-up option.

APPENDIX

The appendixes provide the I/O map, instruction lists, and other information.

- The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU semiconductor device; FUJITSU 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 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 or any third party or does FUJITSU warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU 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 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.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

CONTENTS

CHAPTER 1 OVERVIEW	1
1.1 MB89580B/BW Series Features	2
1.2 MB89580B/BW Series Models	4
1.3 Differences between Models	6
1.4 Block Diagram of MB89580 B/BW Series	8
1.5 Pin Assignments	9
1.6 Package Dimensions	10
1.7 Pin Functions	12
1.8 I/O Circuit Types	16
 CHAPTER 2 HANDLING DEVICE	 19
2.1 Notes on Handling Devices	20
 CHAPTER 3 CENTRAL PROCESSING UNIT (CPU)	 21
3.1 Memory Space	22
3.1.1 Special-Use Areas	24
3.1.2 Location of 16-Bit Data in Memory	26
3.2 Dedicated Registers	28
3.2.1 Condition Code Register (CCR)	30
3.2.2 Register Bank Pointer (RP)	33
3.3 General-Purpose Registers	34
3.4 Interrupts	36
3.4.1 Interrupt Level Registers (ILR1, ILR2, and ILR3)	38
3.4.2 Processing During Interrupt Operation	40
3.4.3 Multiple Interrupts	42
3.4.4 Interrupt Processing Time	43
3.4.5 Stack Operation During Interrupt Processing	44
3.4.6 Stack Area for Interrupt Processing	45
3.5 Resets	46
3.5.1 External Reset Pin	48
3.5.2 Reset Operation	49
3.5.3 State of Pins After a Reset	51
3.6 Clock	52
3.6.1 Clock Generator	54
3.6.2 Clock Controller	55
3.6.3 System Clock Control Register (SYCC)	57
3.6.4 Clock Mode	59
3.6.5 Oscillation Stabilization Wait Time	60
3.7 Standby Mode (Low Power Consumption)	62
3.7.1 Standby Mode Operating State	63
3.7.2 Sleep Mode	64
3.7.3 Stop Mode	65
3.7.4 Standby Control Register (STBC)	66
3.7.5 State Transition Chart	68

3.7.6	Notes on Using Standby Mode	70
3.8	Memory Access Mode	72
CHAPTER 4	I/O PORTS	75
4.1	Overview of the I/O Ports	76
4.2	Port 0	79
4.2.1	Port 0 Registers (PDR0 and DDR0)	81
4.2.2	Operation of Port 0	82
4.3	Port 1	84
4.3.1	Port 1 Registers (PDR1 and DDR1)	86
4.3.2	Operation of Port 1	87
4.4	Port 2	89
4.4.1	Port 2 Register (PDR2)	91
4.4.2	Operation of Port 2	92
4.5	Port 3	93
4.5.1	Port 3 Registers (PDR3 and DDR3)	96
4.5.2	Operation of Port 3	98
4.6	Port 4	100
4.6.1	Port 4 Registers (PDR4 and DDR4)	103
4.6.2	Operation of Port 4	105
4.7	Port 5	107
4.7.1	Port 5 Registers (PDR5 and DDR5)	110
4.7.2	Operation of Port 5	112
4.8	Port 6	114
4.8.1	Port 6 Register (PDR6)	116
4.8.2	Operation of Port 6	117
4.9	Program Example of the I/O Ports	118
CHAPTER 5	TIMEBASE TIMER	119
5.1	Overview of the Timebase Timer	120
5.2	Configuration of the Timebase Timer	122
5.3	Timebase Timer Control Register (TBTC)	124
5.4	Interrupts of the Timebase Timer	126
5.5	Operations of the Timebase Timer	127
5.6	Notes on Using the Timebase Timer	129
5.7	Program Example of the Timebase Timer	130
CHAPTER 6	WATCHDOG TIMER	131
6.1	Overview of the Watchdog Timer	132
6.2	Configuration of the Watchdog Timer	133
6.3	Watchdog Control Register (WDTC)	135
6.4	Operations of the Watchdog Timer	136
6.5	Notes on Using the Watchdog Timer	137
6.6	Program Example of the Watchdog Timer	138
CHAPTER 7	2-CHANNEL 8-BIT PWM TIMER	139
7.1	Overview of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)	140
7.2	Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)	143

7.3	Configuration of the 2-Channel 8-Bit PWM Timer	146
7.4	Pins of the 2-Channel 8-Bit PWM Timer	149
7.5	Registers of the 2-Channel 8-Bit PWM Timer	150
7.5.1	PWM Control Register 1 (CNTR1)	151
7.5.2	PWM Control Register 2 (CNTR2)	153
7.5.3	PWM Control Register 3 (CNTR3)	155
7.5.4	PWM Compare Register 1 (COMR1)	157
7.5.5	PWM compare register 2 (COMR2)	159
7.6	Interrupts of the 2-Channel 8-Bit PWM Timer	161
7.7	Operations of the Interval Timer Function	162
7.8	Operations of the 8-Bit PWM Mode	164
7.9	Operations of the 7-Bit PWM Mode	166
7.10	Operations of the CH12PWM Mode	168
7.11	Prescaler Operation of the 2-Channel 8-Bit PWM Timer	170
7.12	Operation for Standby Mode and Intermediate Stop	172
7.13	Notes on Using the 2-Channel 8-Bit PWM Timer	174
7.14	Program Examples of the 2-Channel 8-Bit PWM Timer (Interval Function)	175
7.15	Program Examples of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)	179
CHAPTER 8	EXTERNAL INTERRUPT CIRCUIT (LEVEL)	183
8.1	Overview of the External Interrupt Circuit (Level)	184
8.2	Configuration of the External Interrupt Circuit	185
8.3	Pins of the External Interrupt Circuit	187
8.4	Registers of the External Interrupt Circuit	190
8.4.1	External Interrupt Control Register (EIE)	191
8.4.2	External Interrupt Flag Register (EIF)	193
8.5	Interrupt of the External Interrupt Circuit	194
8.6	Operations of the External Interrupt Circuit	195
8.7	Program Example of the External Interrupt Circuit	196
CHAPTER 9	PARALLEL PORTS	199
9.1	Overview of the Parallel Ports	200
9.2	Configuration of the Parallel Port	201
9.3	Pins of the Parallel Port	202
9.4	Registers of the Parallel Port	204
9.4.1	Parallel Port Data Register (PDR)	205
9.4.2	Parallel Port Status Register (PSR)	206
9.4.3	Parallel Port Data Control Register (PDCR)	208
9.5	Interrupts of the Parallel Port	210
9.6	Operation of the Parallel Port	211
CHAPTER 10	USB FUNCTION	213
10.1	Overview of the USB Function	214
10.2	Configuration of the USB Function Circuit	216
10.3	Registers of the USB Function Circuit	218
10.3.1	USB Power Mode Register (PMDR)	222
10.3.2	USB Reset Mode Register (UMDR)	223
10.3.3	DMA Base Address Register (DBAR)	225

10.3.4	DMA Base Address Registers for MB89589B (DBARH and DBAR)	227
10.3.5	Transfer Data Count Registers (TDCR0 to TDCR3)	229
10.3.6	USB Control Register (UCTR)	231
10.3.7	USB Status Register 1 (USTR1)	232
10.3.8	USB Status Register 2 (USTR2)	235
10.3.9	USB Interrupt Mask Register (UMSKR)	237
10.3.10	USB Frame Status Register (UFRMR)	240
10.3.11	EndPoint Enable Register (EPER)	241
10.3.12	EndPoint Setup Registers (EPBR0, EPBRx1, EPBR x2)	242
10.4	Interrupts of the USB Function	249
10.5	Functions Supported by the USB Function	251
10.6	Operation of the USB Function	255
10.6.1	Operations of Each Register for Command Responses	258
10.6.2	Suspend Function	262
10.6.3	Wake-up Function	264
CHAPTER 11	UART/SIO	267
11.1	Overview of the UART/SIO	268
11.2	Configuration of the UART/SIO	269
11.3	Pins of the UART/SIO	271
11.4	Registers of the UART/SIO	273
11.4.1	Serial Mode Control Register 1 (SMC1)	274
11.4.2	Serial Mode Control Register 2 (SMC2)	276
11.4.3	Serial Clock Switch Register (SCS)	278
11.4.4	Serial Status And Data Register (SSD)	279
11.4.5	Serial Input Data Register (SIDR)	281
11.4.6	Serial Output Data Register (SODR)	282
11.4.7	Serial Rate Control Register (SRC)	283
11.5	Interrupts of the UART/SIO	284
11.6	Operation of the UART/SIO	285
11.7	Explanation of Operating Mode 0	286
11.8	Explanation of Operating Mode 1	293
CHAPTER 12	CLOCK OUTPUT FUNCTION	299
12.1	Overview of the Clock Output	300
12.2	Clock Output Pin (CLK)	301
12.3	Register of the Clock Output	302
CHAPTER 13	PULL-UP OPTION	303
13.1	Overview of the Pull-up Option	304
13.2	Pull-up Option Setting Register	305
APPENDIX		307
APPENDIX A	I/O Map	308
APPENDIX B	Overview of Instructions	312
B.1	Overview of F ² MC-8L Instructions	313
B.2	Addressing	315
B.3	Special Instructions	319

B.4	Bit Manipulation Instructions (SETB, CLRB)	322
B.5	F ² MC-8L Instructions	323
B.6	Instruction Map	329
APPENDIX C Write Specifications for One-Time PROM and EPROM Microcomputer		330
APPENDIX D Pin State of MB89580B/BW Series		332

CHAPTER 1 OVERVIEW

This chapter explains the features and basic specifications of the MB89580B/BW series of microcontrollers.

- 1.1 "MB89580B/BW Series Features"
- 1.2 "MB89580B/BW Series Models"
- 1.3 "Differences between Models"
- 1.4 "Block Diagram of MB89580B/BW Series"
- 1.5 "Pin Assignments"
- 1.6 "Package Dimensions"
- 1.7 "Pin Functions"
- 1.8 "I/O Circuit Types"

1.1 MB89580B/BW Series Features

The MB89580B/BW series of microcontrollers is a family of general-purpose one-chip microcontrollers. The series features a compact instruction set and incorporates various peripheral functions such as PLL clock control, timers (for example, PWM timer), serial interface, and a USB. The USB function supports a built-in channel and data transfer at both full and low speeds.

■ MB89580B/BW Series Features

○ MB89580B/BW series features

- LQFP64 pin package (0.5 mm pitch), QFP64 pin package (0.65 mm pitch)

○ High-speed, low-voltage operation

Minimum execution time: 0.33 μ s

(The clock frequency supplied externally is 6 MHz. A built-in PLL circuit automatically generates a main clock signal of 12 MHz and a USB interface synchronization clock signal of 48 MHz.)

○ F²MC-8L CPU core

The controller uses an optimal instruction set.

- Multiplication and division instructions
- 16-bit arithmetic operations
- Branch on bit test instructions
- Bit operation instructions

○ PLL clock control

- A built-in PLL clock circuit enables use of an externally supplied low-frequency clock signal that improves noise characteristics. (The externally supplied clock frequency is 6 MHz. The internal system clock frequency is 12 MHz.)

○ Timers

- 8-bit PWM timer (Can be used as an 8-bit PWM timer for two channels or a PPG timer for one channel)
- Built-in 21-bit time-base timer

○ Built-in USB transceiver circuit (supporting transfer at full and low speeds)

○ USB function

- Conformance to USB Protocol Revision 1.0
- Support of transfer at both low and full speeds (selectable)

- Specification of up to four end points
- Supported transfer types: Control, interrupt, bulk, and isochronous
- Built-in DMAC (The buffer at each end point is mapped on built-in RAM, and the RAM is accessed directly to obtain the USB transfer data.)
- **UART and serial interfaces**
 - Built-in UART and ISO functions (selectable)
- **External interrupts**
 - External interrupts (low-level detection for eight interrupt inputs)
Eight interrupt inputs are independent of one other, and can be used to release a device from low power consumption mode. (A low-level detection function is supported.)
- **Low power consumption (standby mode supported)**
 - Stop mode (Oscillation is stopped, and most current consumption is suppressed.)
 - Sleep mode (The CPU is stopped.)
- **Up to 53 general-purpose I/O ports**
 - General-purpose I/O ports (CMOS): 34
 - General-purpose output ports (CMOS): 8
 - General-purpose I/O ports (N-channel open-drain): 3
 - General-purpose input ports (CMOS 3.3 V input supported): 8
- **Parallel ports**
 - Eight general-purpose I/O ports (CMOS) that can also be used as parallel ports
 - Interrupt function supported
 - Asynchronous read/write operations based on external signals
- **Power supply**
 - Voltage: 3.0 V to 5.5 V
- **Package**
 - FTP-64P-M03
Lead pitch: 0.50 mm
Package size: 10 mm x 10 mm
Lead shape: Gull wing
Packaging type: Plastic mold
 - FTP-64P-M09
Lead pitch: 0.65 mm
Package size: 12 mm x 12 mm
Lead shape: Gull wing
Packaging type: Plastic mold

1.2 MB89580B/BW Series Models

The MB89580B series consists of the three models listed in Table 1.2-1 "MB89580B Series Models", and the MB89580BW series consists of the two models listed in Table 1.2-2 "MB89580BW Series Models". Table 1.2-3 "CPU and Peripheral Functions of MB89580B/BW Series Microcontrollers" lists the CPU and peripheral functions.

■ MB89580B/BW Series Models

Table 1.2-1 MB89580B Series Models

	MB9583B	MB89585B	MB89P585B	MB89589B	MB89P589B
ROM size	8KB	16KB	16KB	16KB	16KB
RAM size	512B	1KB	1KB	18KB	18KB
Package	LQFP-64 (FPT-64P-M03)			QFP-64 (FPT-64P-M09)	
Level during a USB reset	High impedance				
Other	Mask ROM product	Mask ROM product	OTP-ROM/ Evaluation product	Mask ROM product	OTP-ROM/ Evaluation product

Table 1.2-2 MB89580BW Series Models

	MB89583BW	MB89585BW	MB89P585BW
ROM size	8KB	16KB	16KB
RAM size	512B	1KB	1KB
Package	LQFP-64 (FPT-64P-M03)		
Level during a USB reset	Low level output		
Other	Mask ROM product	Mask ROM product	OTP-ROM/ Evaluation product

Table 1.2-3 CPU and Peripheral Functions of MB89580B/BW Series

Item			Specification
CPU functions			Number of instructions: 136 Instruction bit length: 8 bits Instruction length: 1 to 3 bytes Data bit length: 1, 8, or 16 bits Minimum execution time: 0.33 μs (at 6 MHz) Interrupt processing time: 3 μs (at 6 MHz)
Peripheral functions	General-purpose ports		General-purpose I/O ports (34 CMOS and 3 N-channel open-drain) General-purpose output ports (8 CMOS) General-purpose input ports (8 CMOS) 3.3 V input
	Parallel ports		The general-purpose I/O ports, eight (P40 to P47) can also be used as parallel ports. Asynchronous read and write operations can be executed with external signals. An interrupt function is available for data setting.
	USB function		Setting of full and low speeds possible Up to 4 EndPoints Built-in FIFO interface (8 x 8 bits) Built-in DMAC (mode setting for DMA transfer to built-in RAM or externally connected FIFO device possible)
	PWM timer		8-bit PWM timer operation x 2 channels (can also be used as a 1-channel PPG timer)
	UART	SIO	UART (data transfer synchronized with clock or asynchronous transfer possible) or SIO (simple serial transfer) selectable
	Time base timer		21-bit time-base timer
	Clock output		Main clock frequency divided by 2 is available for output.
Standby mode			Sleep and stop modes

1.3 Differences between Models

This section notes important points on selecting MB89580 series models.

■ Precautions for Model Selection

○ Memory size

When evaluating a model that uses OTP-ROM, check for the differences between the test model and the production model.

○ Current consumption

- When used at low speed, the model with one-time PROM or EPROM consumes much more current than the model with mask ROM. In sleep or stop mode, the two models consume the same amount of current.
- For details of each package, see Section 1.6 "Package Dimensions".
- For details about power consumption, see the electrical characteristics on the Data Sheet.

○ Differences between the MB89580B series and MB89580BW series

- MB89580B series The pins are kept in the high impedance state until a USB connection is made. Before making a USB connection, use a general-purpose port output to control the circuit with a pull-up resistor connection through software.
- MB89580BW series The pins are kept in the low level until a USB connection is made.

Figure 1.3-1 Connection Example of the MB89580B Series

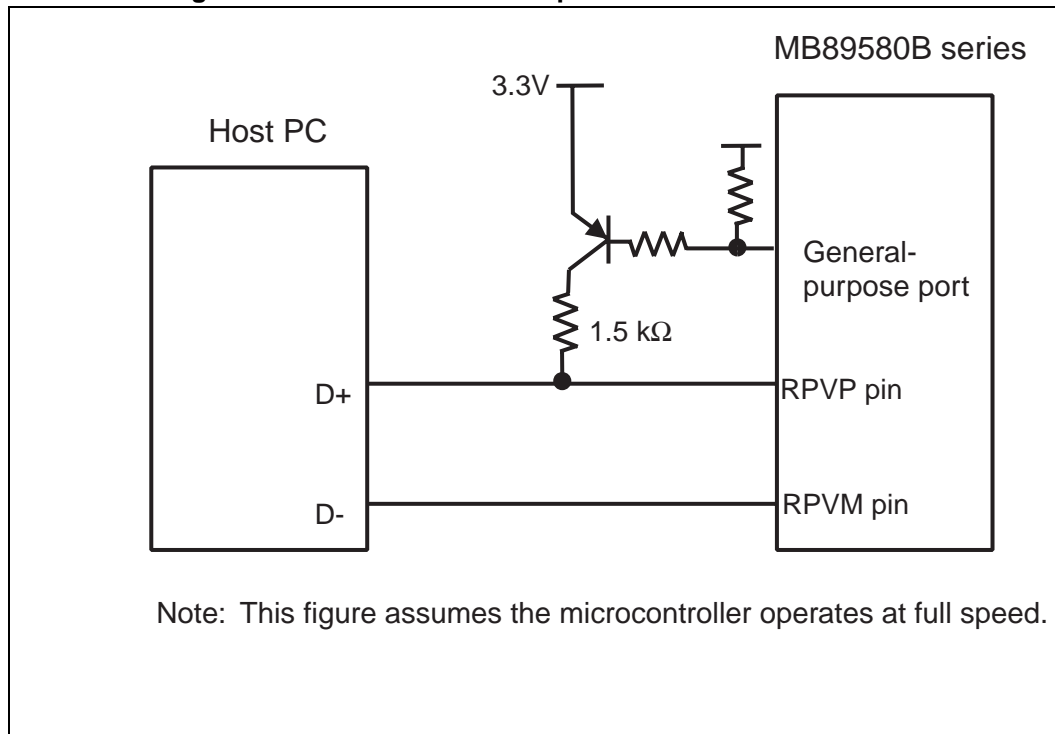
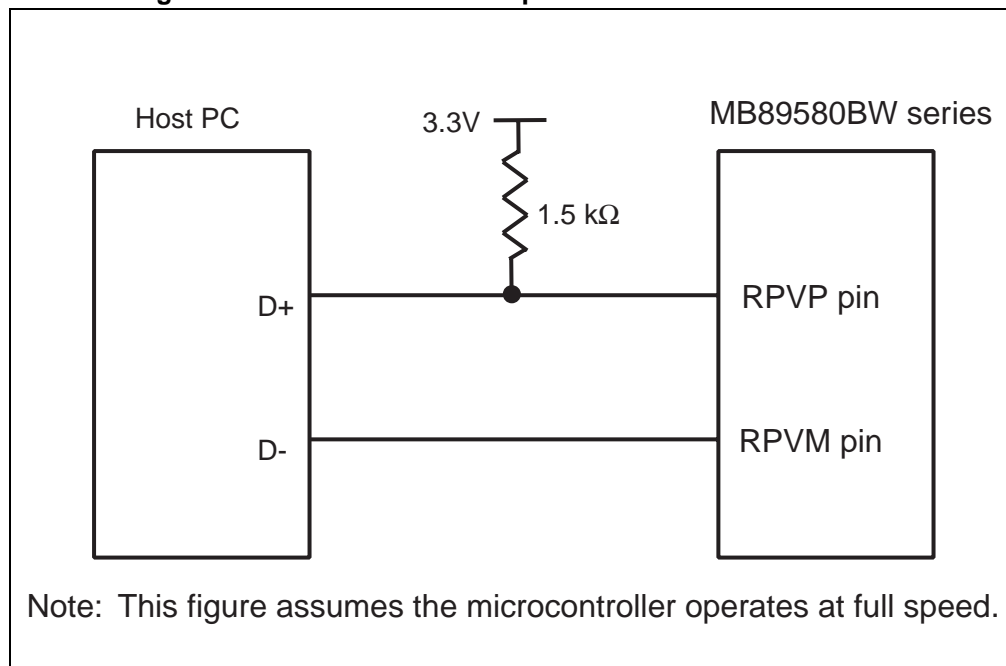


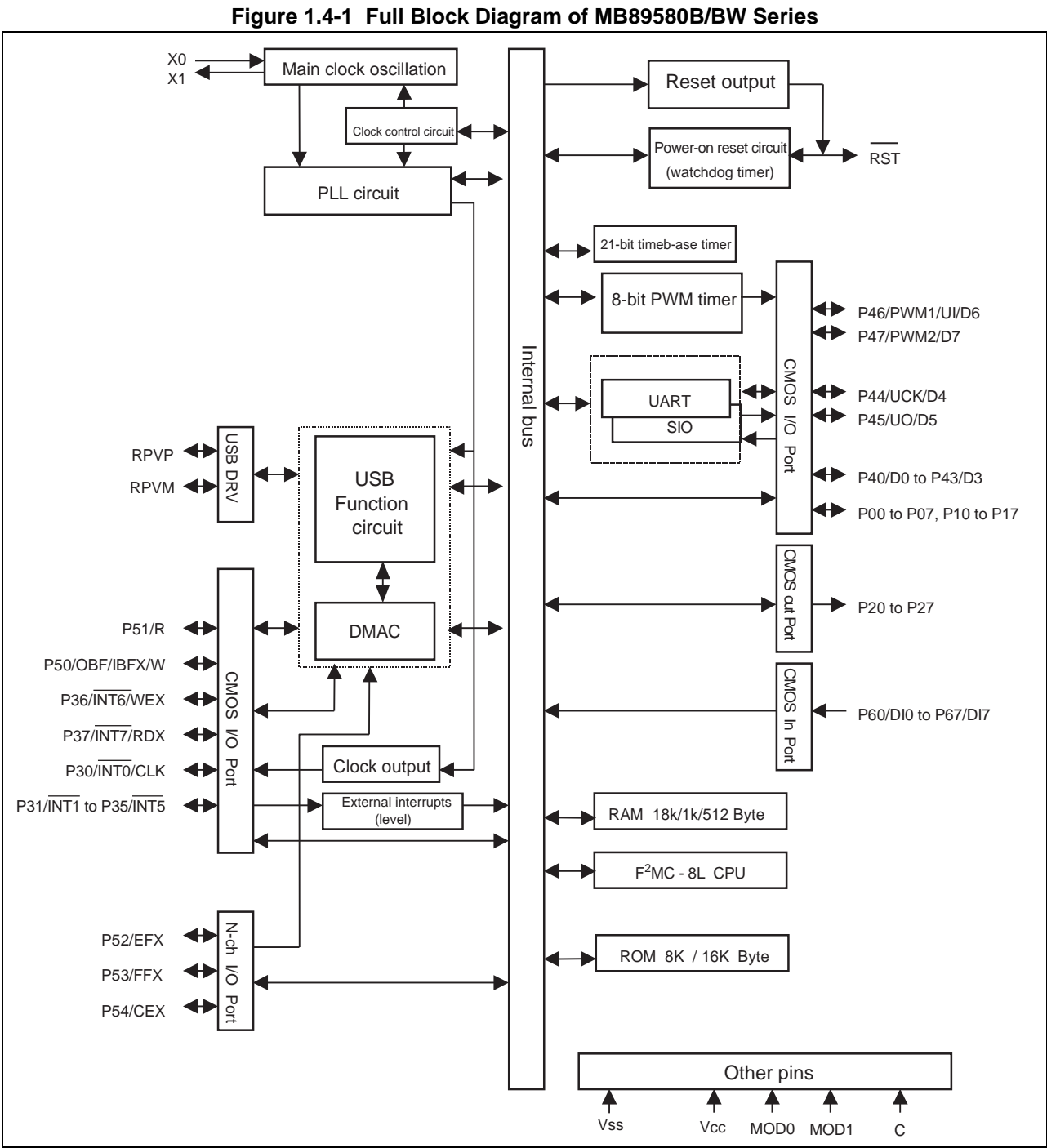
Figure 1.3-2 Connection Example of the MB89580BW Series



1.4 Block Diagram of MB89580 B/BW Series

Figure 1.4-1 "Full Block Diagram of MB89580B/BW Series" is a full block diagram of the MB89580B/BW series.

■ Full Block Diagram of MB89580B/BW Series

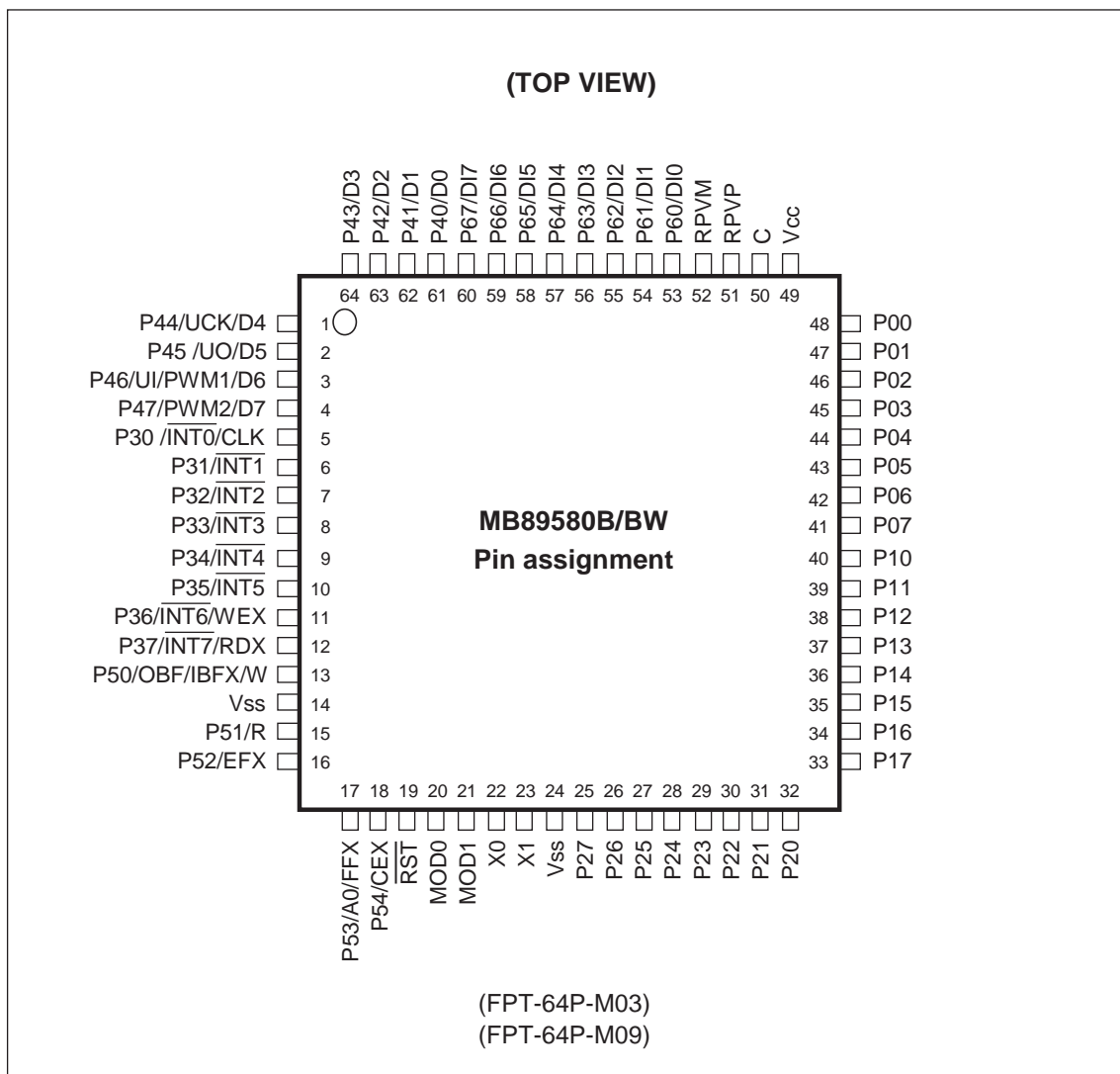


1.5 Pin Assignments

Figure 1.5-1 "MB89580B/BW Series Pin Assignments" shows the MB89580B/BW series pin assignments.

■ MB89580B/BW Series Pin Assignments

Figure 1.5-1 MB89580B/BW Series Pin Assignments



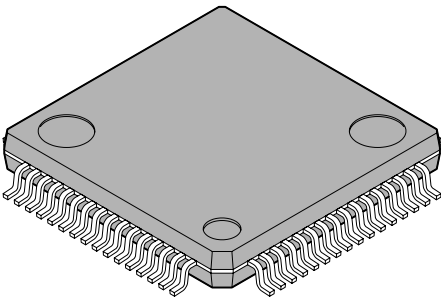
1.6 Package Dimensions

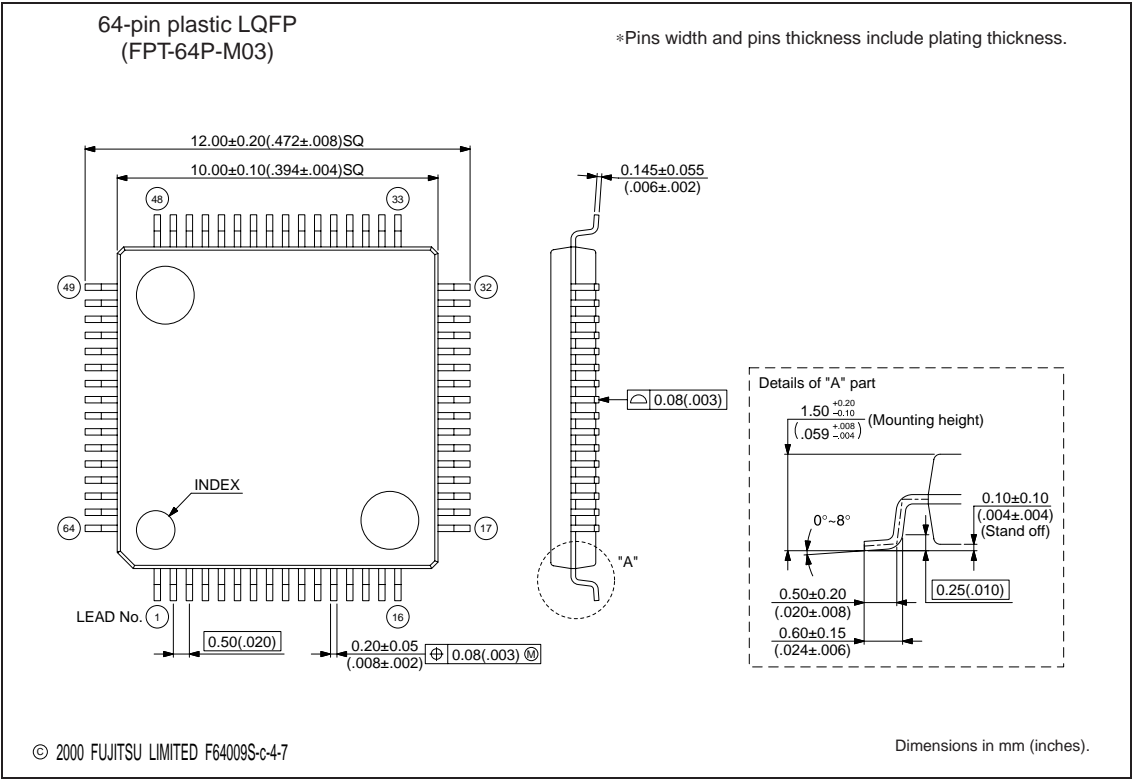
Two types of packages are available in the MB89580B/BW series.

Notes:

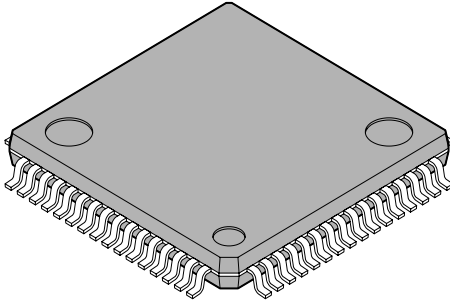
- For MB89589B/P589B, the FPT-64P-M09 only.
- For other models, the FPT-64P-M03 only.

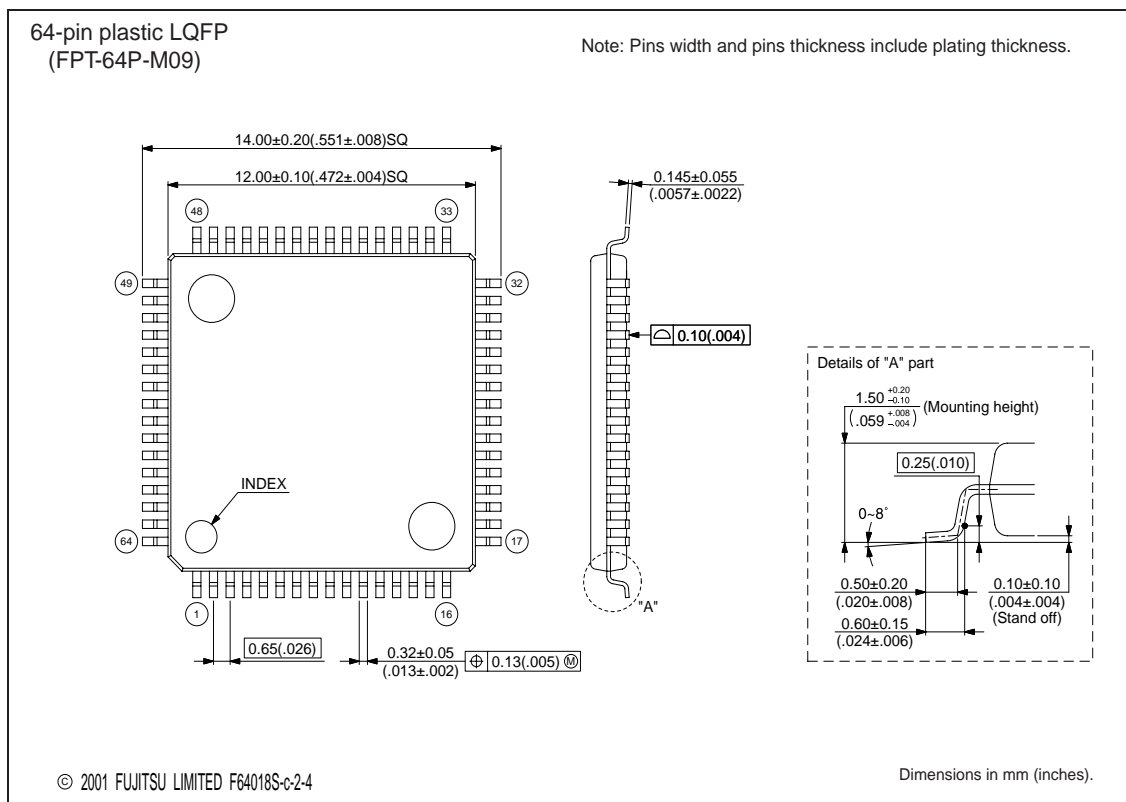
■ Package Dimensions (FPT-64P-M03)

<div>64-pin plastic LQFP</div>  <div>(FPT-64P-M03)</div>	Lead pitch	0.50 mm
	Package width × package length	10.0 × 10.0 mm
	Lead shape	Gull-wing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.32g



■ Package Dimensions (FPT-64P-M09)

<p>64-pin plastic LQFP</p>  <p>(FPT-64P-M09)</p>	Lead pitch	0.65 mm
	Package width × package length	12 × 12 mm
	Lead shape	Gull-wing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX



1.7 Pin Functions

Table 1.7-1 "Pin Functions" lists I/O pins and their functions.

The letters in the "I/O circuit type" column in Table 1.7-1 "Pin Functions" correspond to those in the "Type" column in Table 1.8-1 "I/O Circuit Types".

■ Pin Functions

Table 1.7-1 Pin Functions (1 / 4)

Pin No.	Pin name	I/O circuit type	Description
1	P44/UCK/D4	E	General-purpose CMOS I/O pin UART/SIO clock I/O This pin is also used for parallel interface and external FIFO data output.
2	P45/UO/D5	B	General-purpose CMOS I/O pin UART/SIO serial data output This pin is also used for parallel interface and external FIFO data output.
3	P46/U1/ PWM1/D6	E	General-purpose CMOS input pin UART/SIO serial data input PWM timer This pin is also used for parallel interface and external FIFO data output.
4	P47/PWM2/ D7	B	General-purpose CMOS input pin PWM timer This pin is also used for parallel interface and external FIFO data output.
5	P30/ $\overline{\text{INT0}}$ / CLK	E	General-purpose CMOS I/O pin Clock output This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)
6	P31/ $\overline{\text{INT1}}$	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)
7	P32/ $\overline{\text{INT2}}$	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)
8	P33/ $\overline{\text{INT3}}$	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)

Table 1.7-1 Pin Functions (2 / 4)

Pin No.	Pin name	I/O circuit type	Description
9	P34/ $\overline{\text{INT4}}$	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)
10	P35/ $\overline{\text{INT5}}$	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection)
11	P36/ $\overline{\text{INT6}}$ / WEX	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection) This pin is also used for the input of the parallel interface write strobe.
12	P37/ $\overline{\text{INT7}}$ / RDX	E	General-purpose CMOS I/O pin This pin is also used for external interrupt input. External interrupt input is a hysteresis input. (Level detection) This pin is also used for the input of the parallel interface read strobe.
13	P50/OBF/ IBFX/W	B	General-purpose CMOS I/O pin Host interrupt output from parallel interface This pin is also used for FIFO data strobe output (OUT direction).
14	V _{SS}		Power supply pin (GND)
15	P51/R	B	General-purpose CMOS I/O pin This pin is also used for FIFO data strobe output (IN direction).
16	P52/EFX	K	General-purpose N-channel open-drain I/O pin This pin is also used for FIFO data input enable (IN direction).
17	P53/A0/FFX	K	General-purpose N-channel open-drain I/O pin Data select input to parallel interface This pin is also used for FIFO data input enable (OUT direction).
18	P54/CEX	K	General-purpose N-channel open-drain I/O pin This pin is also used for device select input to the parallel interface.
19	$\overline{\text{RST}}$	I	Reset pin (reset at negative logic low level)
20	MOD0	F	Operating mode select pin Directly connect this pin to V _{SS} .
21	MOD1	F	Operating mode select pin Directly connect this pin to V _{SS} .
22	X0	A	Crystal oscillator pin (6 MHz)
23	X1		
24	V _{SS}	-	Power supply pin (GND)
25	P27	B	General-purpose CMOS output pin
26	P26	B	General-purpose CMOS output pin

CHAPTER 1 OVERVIEW

Table 1.7-1 Pin Functions (3 / 4)

Pin No.	Pin name	I/O circuit type	Description
27	P25	B	General-purpose CMOS output pin
28	P24	B	General-purpose CMOS output pin
29	P23	B	General-purpose CMOS output pin
30	P22	B	General-purpose CMOS output pin
31	P21	B	General-purpose CMOS output pin
32	P20	B	General-purpose CMOS output pin
33	P17	B	General-purpose CMOS I/O pin
34	P16	B	General-purpose CMOS I/O pin
35	P15	B	General-purpose CMOS I/O pin
36	P14	B	General-purpose CMOS I/O pin
37	P13	B	General-purpose CMOS I/O pin
38	P12	B	General-purpose CMOS I/O pin
39	P11	B	General-purpose CMOS I/O pin
40	P10	B	General-purpose CMOS I/O pin
41	P07	B	General-purpose CMOS I/O pin
42	P06	B	General-purpose CMOS I/O pin
43	P05	B	General-purpose CMOS I/O pin
44	P04	B	General-purpose CMOS I/O pin
45	P03	B	General-purpose CMOS I/O pin
46	P02	B	General-purpose CMOS I/O pin
47	P01	B	General-purpose CMOS I/O pin
48	P00	B	General-purpose CMOS I/O pin
49	V _{CC}	-	Power supply pin
50	C	-	Pin for connecting external 0.1 μ F capacitor. When using with 3.3 V power supply, connect this pin with the V _{CC} pin to set to 3.3 V input.
51	RPVP	USBDIV	USB route port + pin
52	RPVM	USBDIV	USB route port - pin
53	P60/DI0	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input (LSB).
54	P61/DI1	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.

Table 1.7-1 Pin Functions (4 / 4)

Pin No.	Pin name	I/O circuit type	Description
55	P62/DI2	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
56	P63/DI3	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
57	P64/DI4	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
58	P65/DI5	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
59	P66/DI6	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
60	P67/DI7	F	General-purpose CMOS input pin (3.3 V input) This pin is also used for external FIFO data input.
61	P40/D0	B	General-purpose CMOS I/O pin This pin is also used for parallel interface and external FIFO data output.
62	P41/D1	B	General-purpose CMOS I/O pin This pin is also used for parallel interface and external FIFO data output.
63	P42/D2	B	General-purpose CMOS I/O pin This pin is also used for parallel interface and external FIFO data output.
64	P43/D3	B	General-purpose CMOS I/O pin This pin is also used for parallel interface and external FIFO data output.

1.8 I/O Circuit Types

Table 1.8-1 "I/O Circuit Types" lists the I/O circuit types. The letters in the "Type" column in Table 1.8-1 "I/O Circuit Types" correspond to those in the "I/O circuit type" column in Table 1.7-1 "Pin Functions".

■ I/O Circuit Types

Table 1.8-1 I/O Circuit Types (1 / 2)

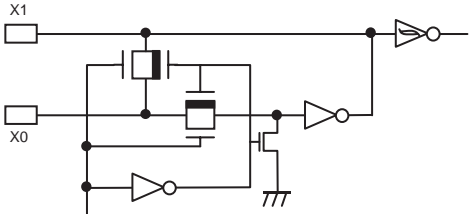
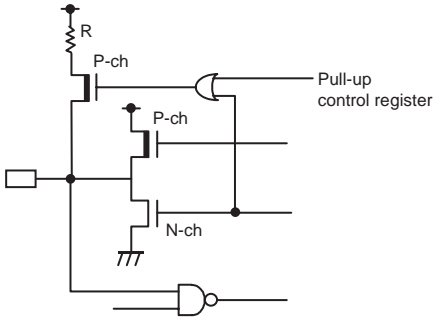
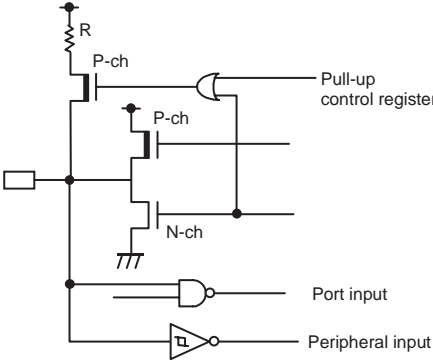

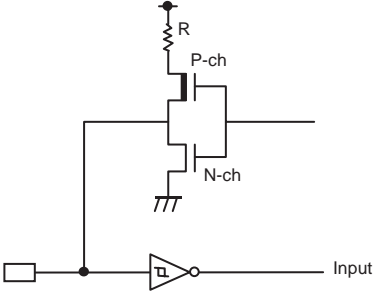
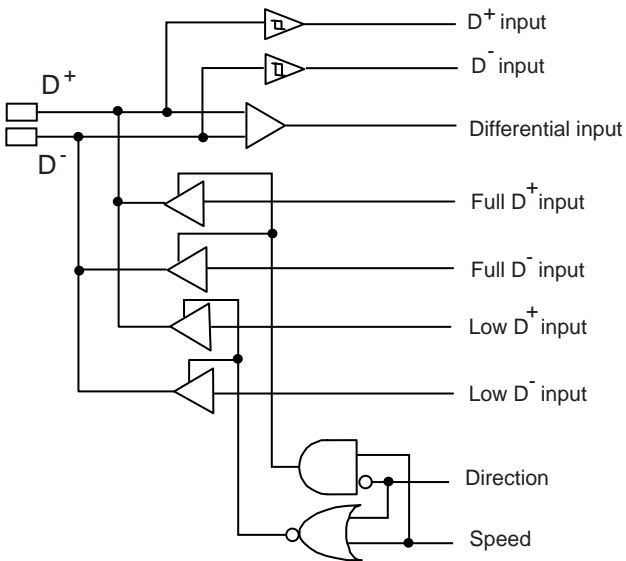
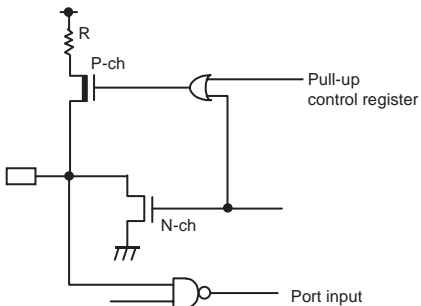
Type	Circuit	Remarks
A		Oscillation feedback resistor About 1 MΩ
B		• CMOS I/O
E		• CMOS I/O • Hysteresis input

Table 1.8-1 I/O Circuit Types (2 / 2)

Type	Circuit	Remarks
F		<ul style="list-style-type: none"> CMOS I/O
I		<ul style="list-style-type: none"> Hysteresis I/O Pull-up resistor
USBDV		<ul style="list-style-type: none"> Hysteresis input N-channel open-drain I/O
K		<ul style="list-style-type: none"> N-channel open-drain I/O

CHAPTER 2 HANDLING DEVICE

This chapter explains the precautions to be taken when using the USB general-purpose one-chip microcontroller.

2.1 "Notes on Handling Devices"

2.1 Notes on Handling Devices

This section explains the precautions to be taken when handling the power supply voltage and pins of the device.

■ Notes on Handling Devices

○ Preventing latch-up

latch-up may occur on CMOS ICs if voltage higher than V_{CC} or lower than V_{SS} is applied to input or output pins other than the medium- and high-voltage pins or if voltage higher than the rating is applied between V_{CC} and V_{SS} .

When latch-up occurs, power supply current increases rapidly and might thermally damage elements. When using, take great care not to exceed the absolute maximum ratings.

○ Power supply voltage fluctuations

Although V_{CC} power supply voltage is assured to operate within the rated range, a rapid fluctuation of the voltage could cause malfunctions even if it occurs within the rated range. Stabilizing voltage supplied to the IC is therefore important.

As stabilization guidelines, it is recommended to control power so that V_{CC} ripple fluctuations (P-P value) will be less than 10% of the standard V_{CC} value at the commercial frequency (50 to 60 Hz) and the transient fluctuation rate will be less than 0.1 V/ms at the time of a momentary fluctuation such as when power is switched.

○ Treatment of unused input pins

Leaving unused input pins open could cause malfunctions and latch-up leading to permanent damage to the pins. These unused pins should be connected to a pull-up or pull-down resistance of at least 2 k Ω between the pin and the power supply.

Unused I/O pins should be placed in output state to leave it open or pins that are in input state should be handled the same as unused input pins.

○ Note on operations during PLL clock mode

If the PLL clock mode is selected, the microcontroller attempt to be working with the self-oscillating circuit even when there is no external oscillator or external clock input is stopped. Performance of this operation, however, cannot be guaranteed.

CHAPTER 3 CENTRAL PROCESSING UNIT (CPU)

This chapter explains the functions and operation of the CPU.

- 3.1 "Memory Space"
- 3.2 "Dedicated Registers"
- 3.3 "General-Purpose Registers"
- 3.4 "Interrupts"
- 3.5 "Resets"
- 3.6 "Clock"
- 3.7 "Standby Mode (Low Power Consumption)"
- 3.8 "Memory Access Mode"

3.1 Memory Space

The MB89580B/BW memory space is 64K bytes and consists of an I/O area, RAM area, and ROM area. The memory space also includes areas, such as general-purpose registers and vector tables, that are used for specific applications.

■ Memory Space Configuration

○ I/O area (address: 0000_H to 007F_H)

- The control registers and data registers of built-in peripheral functions are allocated.
- The I/O area is allocated to a part of the memory space, and can be accessed in the same way as memory. Direct addressing enables faster access to the I/O area.

○ RAM area

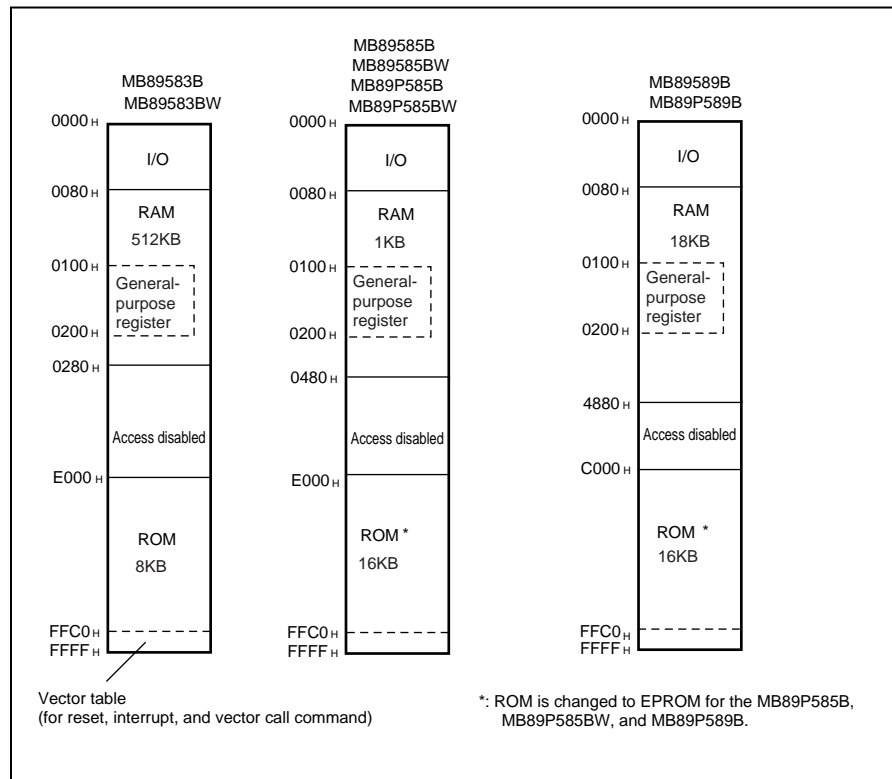
- Static RAM is included for the internal data area.
- The internal RAM size depends on the model.
- Direct addressing enables fast access to the area from 80_H to FF_H.
- The area from 100_H to 1FF_H can be used as a general-purpose register area (the amount of usable area depends on the model).
- A reset causes RAM data to be undefined.

○ ROM area

- ROM is included for the internal program area.
- The internal ROM size depends on the model.
- The area from FFC0_H to FFFF_H is used for the vector table.

○ Memory map

Figure 3.1-1 Memory Map



3.1.1 Special-Use Areas

In addition to the I/O area, the general-purpose register and vector table areas are available as special-use areas.

■ General-purpose Register Area (address: 0100_H to 01FF_H)

- This area provides a auxiliary registers that can be used for 8-bit arithmetic operations and transfer.
- This area is allocated to a part of the RAM area, and can also be used as normal RAM.
- When this area is used as general-purpose registers, general-purpose register addressing enables fast access to this area with a short instruction.

See Section 3.2.2 "Register Bank Pointer (RP)", and Section 3.3 "General-Purpose Registers", for details.

■ Vector Table Area (address: FFC0_H to FFFF_H)

- This area is used as vector tables for the vector call instruction, interrupts, and resets.
- This area is allocated at the beginning of the ROM area. The start addresses of the corresponding processing routines are set as the data for the vector tables addresses.

Table 3.1-1 "Vector Tables" lists the addresses of the vector tables that are referenced by the vector call instruction, interrupts, and resets.

See Section 3.4 "Interrupts", Section 3.5 "Reset", and (6) CALLV #vct in Appendix B.3 "Special Instructions", for details.

Table 3.1-1 Vector Tables

Vector call instruction	Addresses of vector table	
	Upper	Lower
CALLV #0	FFC0 _H	FFC1 _H
CALLV #1	FFC2 _H	FFC3 _H
CALLV #2	FFC4 _H	FFC5 _H
CALLV #3	FFC6 _H	FFC7 _H
CALLV #4	FFC8 _H	FFC9 _H
CALLV #5	FFCA _H	FFCB _H
CALLV #6	FFCC _H	FFCD _H
CALLV #7	FFCE _H	FFCF _H

Interrupts	Addresses of vector table	
	Upper	Lower
IRQB	FFE4 _H	FFE5 _H
IRQA	FFE6 _H	FFE7 _H
IRQ9	FFE8 _H	FFE9 _H
IRQ8	FFEA _H	FFEB _H
IRQ7	FFEC _H	FFED _H
IRQ6	FFEE _H	FFEF _H
IRQ5	FFF0 _H	FFF1 _H
IRQ4	FFF2 _H	FFF3 _H
IRQ3	FFF4 _H	FFF5 _H
IRQ2	FFF6 _H	FFF7 _H
IRQ1	FFF8 _H	FFF9 _H
IRQ0	FFFA _H	FFFB _H
Mode data	- *	FFFD _H
Reset vector	FFFE _H	FFFF _H

*: FFFC_H is not available (FF_H is set).

3.1.2 Location of 16-Bit Data in Memory

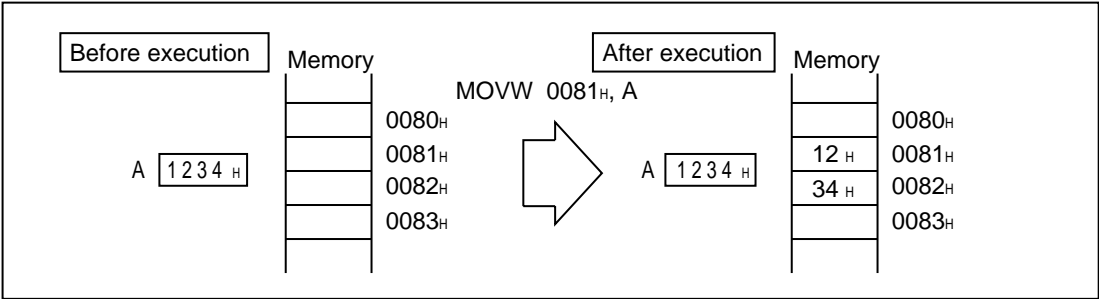
The high-order data of 16-bit data and stack data is stored at the lower address in memory.

■ Storage State of 16-Bit Data in RAM

The high-order byte of 16-bit data written to memory is stored at the lower address. The low-order byte of the data is stored at the next address. Data read from memory is also handled the same way.

Figure 3.1-2 "Location of 16-Bit Data in Memory" shows the storage location of 16-bit data in memory.

Figure 3.1-2 Location of 16-Bit Data in Memory



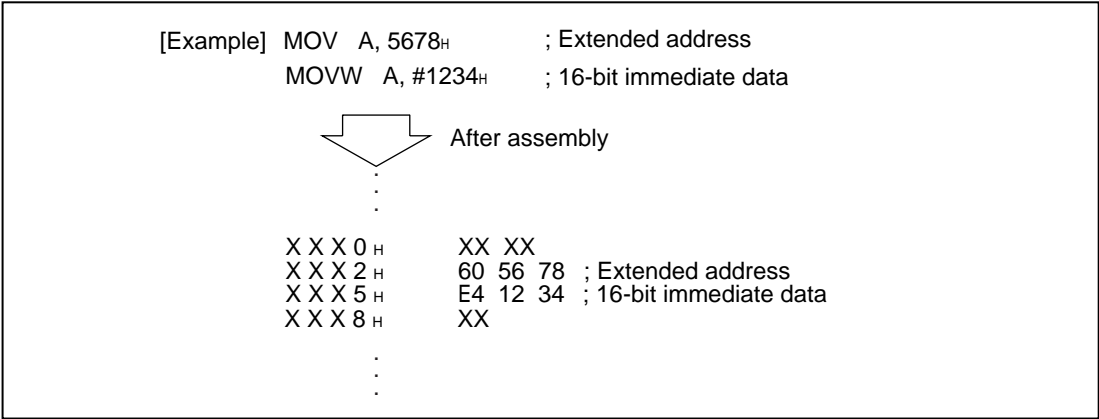
■ Storage State of a 16-Bit Operand

If 16-bit data is specified by the operand in an instruction, the high-order byte is also stored at the address nearer the operation code (instruction). The low-order byte is also stored at the next address.

This is also true both if the operand indicates a memory address or if it indicates 16-bit immediate data.

Figure 3.1-3 "Location of 16-Bit Data in an Instruction" shows the location of 16-bit data in an instruction.

Figure 3.1-3 Location of 16-Bit Data in an Instruction



■ Storage State of 16-Bit Stack Data

The high-order byte of 16-bit register data saved on the stack by an interrupt is also stored at the lower address in the same way.

3.2 Dedicated Registers

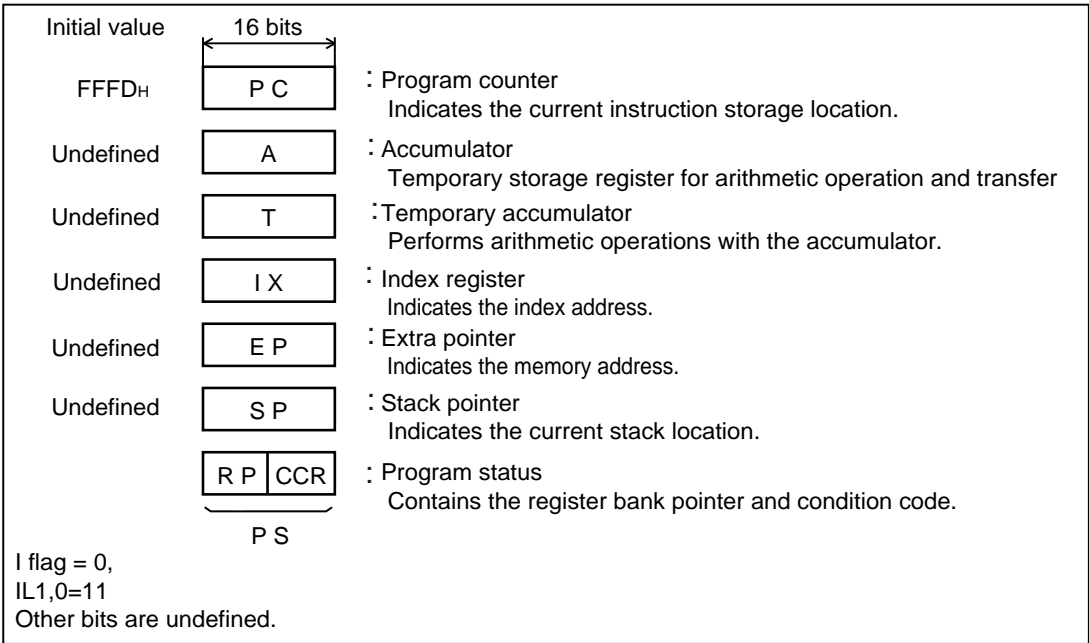
The dedicated registers in the CPU include the program counter (PC), two arithmetic operation registers (A and T), three address pointers (IX, EP, and SP), and program status (PS). Each register is 16 bits.

■ Dedicated Register Configuration

The dedicated registers in the CPU include seven 16-bit registers. In some of the registers, only the low-order 8 bits can be used.

Figure 3.2-1 "Dedicated Register Configuration" shows the dedicated register configuration.

Figure 3.2-1 Dedicated Register Configuration



■ Dedicated Register Functions

○ Program counter (PC)

The program counter is 16 bits and indicates the memory address of the instruction the CPU is currently being executed. Instruction execution, interrupts, and resets update the contents of the program counter. During reset operation, the initial value is the address (FFFDH) at which the mode data is read.

○ **Accumulator (A)**

The accumulator is a 16-bit arithmetic operation register and uses memory data and data in another register, such as the temporary accumulator (T), for arithmetic operations and transfer. Data in the accumulator can also be handled as both word length (16 bits) and byte length (8 bits). When arithmetic operations and transfer are performed in byte length, only low-order 8 bits (AL) of the accumulator are used. The high-order 8 bits (AH) remain unchanged. After a reset, the initial value is undefined.

○ **Temporary accumulator (T)**

The temporary accumulator is a 16-bit auxiliary arithmetic operation register that is used to perform arithmetic operations on data in the accumulator (A). Data in the temporary accumulator is handled as word length if arithmetic operations for the accumulator (A) are word length (16 bits). The data is handled as byte length if arithmetic operations are byte length (8 bits). When byte-length arithmetic operations are performed, only the low-order 8 bits (TL) of the temporary accumulator are used. The high-order 8 bits (TH) are not used.

When a MOV instruction is used to transfer data to the accumulator (A), data in the accumulator before transfer is automatically transferred to the temporary accumulator. For a byte-length transfer, the high-order 8 bits (TH) of the temporary accumulator are not changed. After a reset, the initial value is undefined.

○ **Index register (IX)**

The index register is 16 bits, and holds the index address. When used, the index register is combined with a one-byte offset (-128 to +127). It generates a memory address for accessing data by adding the code-expanded offset value to the index address. After a reset, the initial value is undefined.

○ **Extra pointer (EP)**

The extra pointer is a 16-bit register that provides the memory address for accessing data. After a reset, the initial value is undefined.

○ **Stack pointer (SP)**

The stack pointer is a 16-bit register that hold the addresses referenced by interrupts, subroutine calls, and the stack save and restore instructions. The value of the stack pointer during program execution is the address containing the latest data saved on the stack. After a reset, the initial value is undefined.

○ **Program status (PS)**

The program status register is a 16-bit control register. The high-order 8 bits are the register bank pointer (RP), which is used to indicate the address of a general-purpose register.

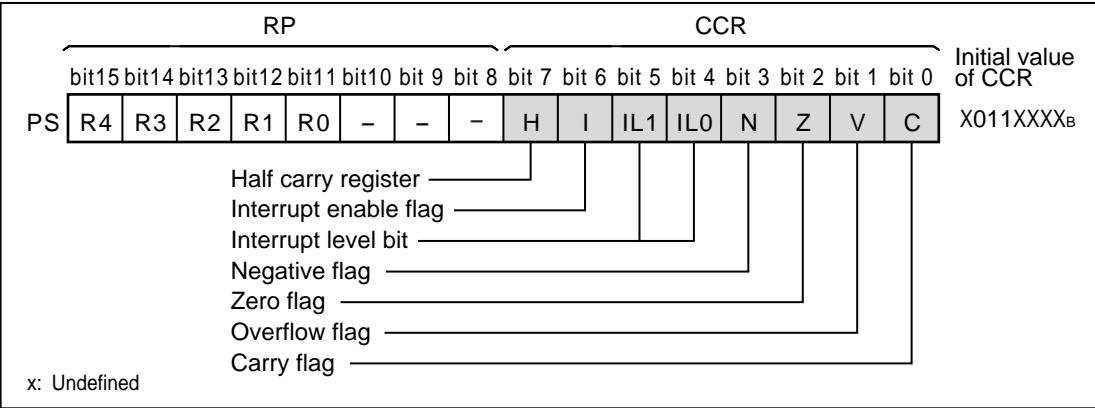
The low-order 8 bits are the condition code register (CCR), which consists of flags that indicate the CPU state. Each 8-bit register forms part of the program status, and cannot be accessed independently (the only instructions that access the program status are MOVW A, PS and MOVW PS, A).

3.2.1 Condition Code Register (CCR)

The condition code register (CCR) is the low-order 8 bits of the program status (PS) register. This register consists of the bits (C, V, Z, N, and H) that indicate the arithmetic operation results and the transfer data contents and the bits (I, IL1, and IL0) that control acceptance of an interrupt request.

■ Configuration of Condition Code Register (CCR)

Figure 3.2-2 Configuration of Condition Code Register



■ Bits That Indicate Arithmetic Operation Results

○ Half carry flag (H)

As a result of an arithmetic operation, this flag is set to 1 when a carry from bit 3 to bit 4 or a borrow from bit 4 to bit 3 occurs. In other cases, this flag is cleared to 0. This flag is intended for decimal adjustment instructions. Use this flag only for addition and subtraction.

○ Negative flag (N)

As a result of an arithmetic operation, this flag is set to 1 when the highest bit is set to 1, and is cleared to 0 when it is set to 0.

○ Zero flag (Z)

This flag is set to 1 if the result of an arithmetic operation is 0. In other cases, this flag is cleared to 0.

○ Overflow flag (V)

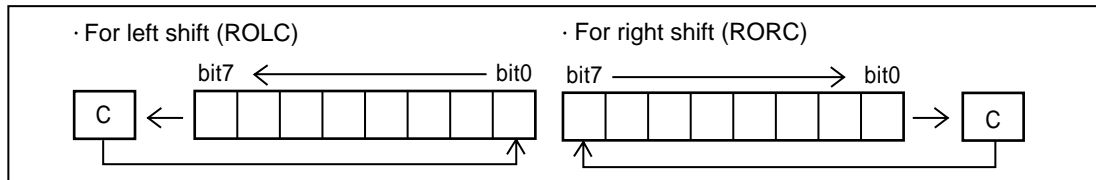
As a result of an arithmetic operation, this flag is set to 1 when two's complement overflow occur. This flag is cleared to 0 when the overflow does not occur.

○ Carry flag (C)

As a result of an arithmetic operation, this flag is set to 1 when a carry from bit 1 or a borrow to bit 7 occurs. This flag is cleared to 0 otherwise. In a shift instruction, this flag becomes the shift-out value.

Figure 3.2-3 "Carry Flag Change Caused by Shift Instruction" shows the change in the carry flag caused by the shift instruction.

Figure 3.2-3 Carry Flag Change Caused by Shift Instruction



Note:

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

Reference:

The flags are hardly ever used directly after the flag bits are fetched. Usually, the flags are used indirectly by the branch instruction (BNZ) and decimal adjustment instructions (DAA and DAS). The initial values of these flags are undefined after a reset.

■ Bits That Control Acceptance of an Interrupt

○ Interrupt enable flag (I)

When this flag is set to 1, interrupts are enabled and the CPU accepts them. When this flag is set to 0, interrupts are disabled and the CPU does not accept them.

After a reset, the initial value is set to 0.

Usually, this flag is set to 1 by the SETI instruction and cleared to 0 by the CLRI instruction.

○ Interrupt level bits (IL1 and IL0)

These bits indicate the level of interrupts that the CPU currently accepts, and are compared with the value of the interrupt level register (ILR1 to ILR3) set according to interrupt requests (IRQ0 to IRQB) from the peripheral functions.

The CPU processes an interrupt only if the interrupt request has an interrupt level that is smaller than the values indicated by these bits when the interrupt enable flag is enabled (I = 1). Table 3.2-1 "Interrupt Levels" lists priority of interrupt levels. After a reset, the initial value is set to 11.

Table 3.2-1 Interrupt Levels

IL1	IL0	Interrupt level	Priority
0	0	1	High ↑ ↓ Low (no interrupt)
0	1		
1	0	2	
1	1	3	

Reference:

While the CPU is not processing an interrupt (during execution of main program), the interrupt level bits (IL1 and IL0) are set to 11.

See Section 3.4 "Interrupts", for more information about interrupts.

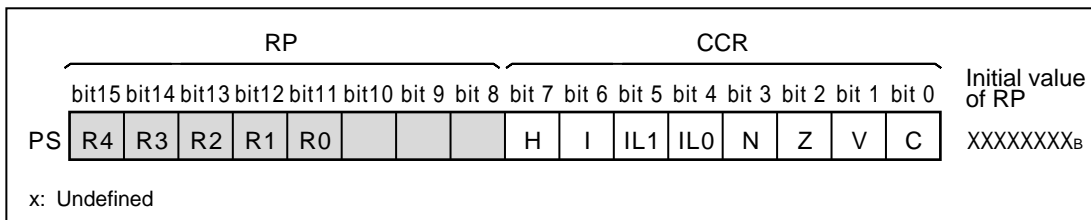
3.2.2 Register Bank Pointer (RP)

The register bank pointer (RP), which is the high-order 8 bits of the program status (PS), indicates the address of the general-purpose register bank currently in use. The register bank pointer is converted to an actual address in general-purpose register addressing mode.

■ Configuration of Register Bank Pointer (RP)

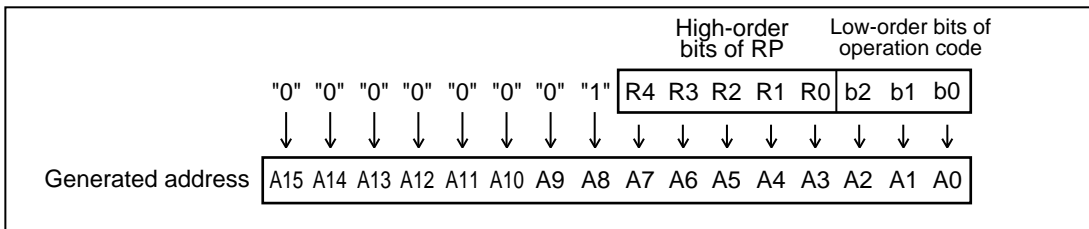
Figure 3.2-4 "Configuration of Register Bank Pointer" shows the configuration of register bank pointer.

Figure 3.2-4 Configuration of Register Bank Pointer



The register bank pointer indicates the address of the register bank currently in use. The relationship between the contents of the register bank pointer and the actual address follows the conversion rule shown in Figure 3.2-5 "Conversion Rule for Actual Address in General-purpose Register Area".

Figure 3.2-5 Conversion Rule for Actual Address in General-purpose Register Area



The register bank pointer specifies the memory block (register bank) that is to be used as the general-purpose register in the RAM area. There are 32 register banks in all, which are specified by setting a value of 0 to 31 in the high-order 5 bits of the register bank pointer. One register bank has eight 8-bit general-purpose registers that are selected by the low-order 3 bits of the operation code.

The register bank pointer enables the area from 0100H to 01FFH to be used as the general-purpose register area. The amount of usable area depends on the model. After a reset, the initial value is undefined.

Note:

Before using a general-purpose register, always set the register bank pointer (RP).

The register bank pointer is part of the program status (PS), and therefore cannot be accessed independently.

3.3 General-Purpose Registers

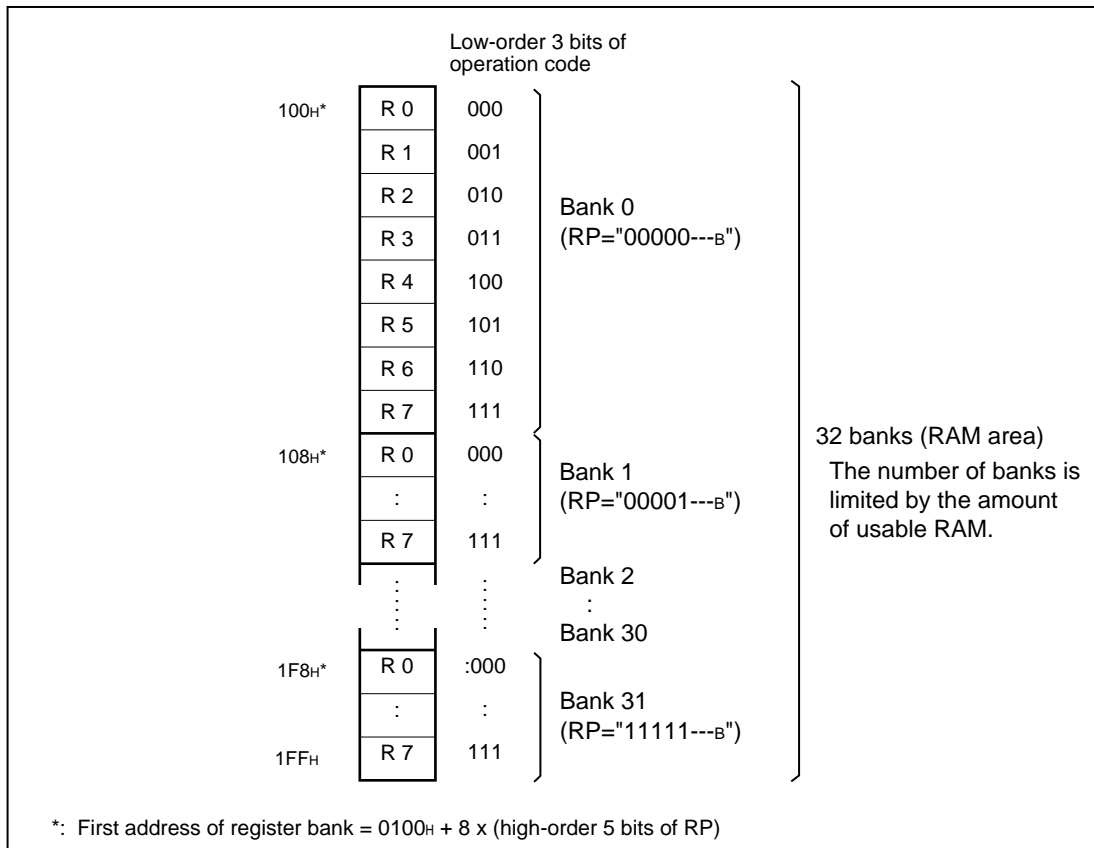
The general-purpose registers are a memory block in which each bank is 8 bits x 8. The register bank pointer (RP) is used to specify a register bank. Functionally, up to 32 banks in all can be used, but the internal RAM size is small and all banks may not be able to be used. Using the general-purpose registers for interrupt processing, vector call processing, and subroutine call is effective.

■ General-Purpose Register Configuration

- A general-purpose register is 8 bits and is located in a register bank in the general-purpose register area (RAM).
- Each bank has eight registers (R0 to R7), and up to 32 banks in all can be used. However, if internal RAM alone is used, the number of banks that can be used depends on the model.
- The register bank currently in use is specified by the register bank pointer (RP).

Figure 3.3-1 "Register Bank Configuration" shows the register bank configuration.

Figure 3.3-1 Register Bank Configuration



See Section 3.1.1 "Special-Use Areas", for information about the general-purpose register area that can be used for each model.

■ Features of General-Purpose Registers

General-purpose registers have the following features:

- A short instruction can be used for fast RAM access (general-purpose register addressing).
- The general-purpose registers are divided into blocks of register banks to facilitate the protection of register contents and to facilitate separation into functional units.

General-purpose registers enable permanent allocation of dedicated register banks to the interrupt processing routines and vector call (CALLV #0 to #7) processing routines. For example, the fourth register bank is used for the second interrupt.

Unless another routine accidentally rewrites the contents of the dedicated register bank used for the processing of an interrupt, the contents of general-purpose registers existing before an interrupt are only saved by specifying the dedicated register bank at the beginning of the interrupt processing routine. This eliminates the need to save the general-purpose registers on the stack and enables the fast acceptance of an interrupt without any confusion being caused.

For subroutine calls, in addition to protection of the contents of general-purpose registers, the reentrant program (reentrant program that does not use a fixed variable address) usually created when the index register (IX) is used can also be implemented with a register bank.

Note:

An interrupt processing routine must be programmed so that the values of interrupt level bits (CCR: IL1 and IL0) of the condition code register are not changed when the register bank pointer (RP) is rewritten to specify the register bank.

3.4 Interrupts

MB89580B/BW series products have 12 interrupt request inputs for peripheral functions. Interrupt levels can be set independently for each. The interrupt controller compares the interrupt levels of interrupt requests issued from peripheral functions if output of interrupt requests from peripheral functions is enabled. The CPU performs interrupt operations according to the interrupt acceptance state. An interrupt request releases standby mode, and returns control to interrupt operation or normal operation.

■ **Interrupt Requests from Peripheral Functions**

Table 3.4-1 "Functions of TBTC Bits" lists the interrupt requests for the peripheral functions. When an interrupt is accepted, the system uses the contents of the interrupt vector table address corresponding to the interrupt request as the branch destination address to branch to the interrupt processing routine.

Interrupt requests enable interrupt processing to be set to three priority levels with the interrupt level registers (ILR1, ILR2, and ILR3).

Interrupt requests at the same or lower level occurring during execution of a interrupt processing routine are usually processed after the current interrupt processing routine is terminated. If interrupt requests of the same level occur concurrently, IRQ0 has the highest priority.

Table 3.4-1 "Functions of TBTC Bits" Interrupt requests and interrupt vector

Table 3.4-1 Functions of TBTC Bits

Interrupt request	Address of vector table		Name of interrupt level register bit	Priority when the same level occurs concurrently
	Higher	Lower		
IRQ0 (external interrupt)	FFFA _H	FFFB _H	L01,L00	<div>High</div> <div>↑</div> <div>↓</div> <div>Low</div>
IRQ1 (parallel port)	FFF8 _H	FFF9 _H	L11,L10	
(Unused)	FFF6 _H	FFF7 _H	L21,L20	
IRQ3 (USB FUNCTION 1)	FFF4 _H	FFF5 _H	L31,L30	
(Unused)	FFF2 _H	FFF3 _H	L41,L40	
IRQ5 (USB FUNCTION 2)	FFF0 _H	FFF1 _H	L51,L50	
IRQ6 (UART/SIO)	FFFE _H	FFFF _H	L61,L60	
IRQ7 (timebase timer)	FFFC _H	FFFD _H	L71,L70	
IRQ8 (8-bit PWM timers 1 and 2)	FFFA _H	FFFB _H	L81,L80	
IRQ9 (unused)	FFF8 _H	FFF9 _H	L91,L90	
IRQA (unused)	FFF6 _H	FFF7 _H	LA1,LA0	
IRQB (unused)	FFF4 _H	FFF5 _H	LB1,LB0	

3.4.1 Interrupt Level Registers (ILR1, ILR2, and ILR3)

Twelve 2-bit data units corresponding to interrupt requests from the peripheral functions are assigned to the interrupt level registers (ILR1, ILR2, and ILR3). Interrupt levels can be set in these 2-bit data (interrupt level bits).

■ Configuration of Interrupt Level Registers (ILR1, ILR2, and ILR3)

Figure 3.4-1 Configuration of Interrupt Level Register

Register	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
ILR1	0 0 7 C _H	L31	L30	L21	L20	L11	L10	L01	L00	11111111 _B
		W	W	W	W	W	W	W	W	
ILR2	0 0 7 D _H	L71	L70	L61	L60	L51	L50	L41	L40	11111111 _B
		W	W	W	W	W	W	W	W	
ILR3	0 0 7 E _H	LB1	LB0	LA1	LA0	L91	L90	L81	L80	11111111 _B
		W	W	W	W	W	W	W	W	

W:Write-only

Two bits in an interrupt level register are assigned to each interrupt request. The values of interrupt level bits set in the registers provide priority levels for interrupt processing (interrupt levels 1 to 3).

The interrupt level bits are compared with the interrupt level bits (CCR: IL1 and IL0) of the condition code register.

If interrupt level 3 is set, the CPU does not accept any interrupts.

Table 3.4-2 "Relationship Between Interrupt Level Bits and Interrupt Levels" shows the relationship between the interrupt level bits and the interrupt levels.

Table 3.4-2 Relationship Between Interrupt Level Bits and Interrupt Levels

L01-LB1	L00-LB0	Interrupt level	Priority
0	0	1	High ↕ Low (no interrupt)
0	1		
1	0	2	
1	1	3	

Reference:

During execution of the main program, the interrupt level bits (CCR: IL1 and IL0) of the condition code register are usually set to 11.

Note:

Because ILR1, ILR2, and ILR3 registers are write-only, the bit manipulation instructions (SETB and CLRB) cannot be used.

3.4.2 Processing During Interrupt Operation

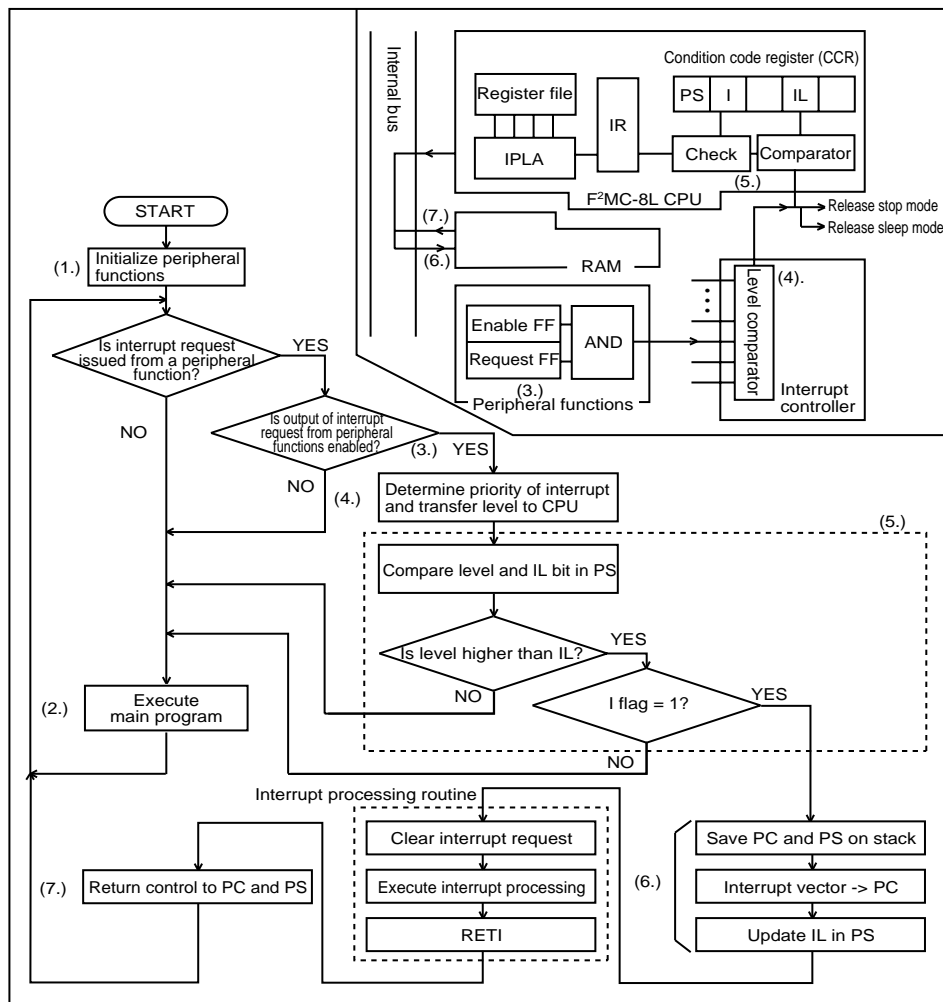
When an interrupt request is issued from a peripheral function, the interrupt controller sends the interrupt level to the CPU. If the CPU is ready to accept an interrupt, it temporarily stops the current program and executes the interrupt processing routine.

■ Processing During Interrupt Operation

The steps during interrupt operation are the occurrence of an interrupt cause for a peripheral function, setting of the interrupt request flag bit (request FF), determination of the interrupt request enable bit (enable FF), determination of the interrupt level (ILR1, ILR2, ILR3, and CCR: IL1 and IL0), determination of whether concurrent same-level requests exist, and determination of the interrupt enable flag (CCR: I).

Figure 3.4-2 "Processing During Interrupt Operation" shows the processing during interrupt operation.

Figure 3.4-2 Processing During Interrupt Operation



1. After a reset, all interrupt requests are disabled.

The peripheral function initialization program is used to initialize peripheral functions that generate interrupts. The interrupt levels are set in the interrupt level registers (ILR1, ILR2, and ILR3), then peripheral functions start operation.

Interrupt level 1, 2, or 3 can be set. Level 1 is the highest, and level 2 is the second highest. If level 3 is set, interrupts from the relevant peripheral functions are disabled.

2. The main program (for multiple interrupts, the interrupt processing routine) is executed.
3. When an interrupt cause occurs in a peripheral function, the interrupt request flag bit (request FF) of the peripheral function is set to 1. When the interrupt request enable bit of the peripheral function is set to enable (enable FF = 1), the interrupt request is output to the interrupt controller.
4. The interrupt controller always monitors the interrupt requests from peripheral functions, and transfers the highest interrupt level of the current interrupt requests to the CPU. If requests with the same interrupt level are issued concurrently, priority is also determined.
5. If the received interrupt level has higher priority (level is a smaller number) than the level set in the interrupt level bit (CCR: IL1 and IL0) of the condition code register, the CPU checks the contents of the interrupt enable flag (CCR: I). The CPU accepts an interrupt if the flag is set to enable (CCR: I = 1).
6. The contents of the program counter (PC) and the program status (PS) are saved on the stack. The first address of the interrupt processing routine is fetched from the appropriate interrupt vector table. The values of the interrupt level bits (CCR: IL1 and IL0) of the condition code register are changed to the value of the accepted interrupt level. Execution of the interrupt processing routine is then started.
7. The RETI instruction is used to restore the program counter (PC) and program status (PS) values. Processing is executed from the instruction that follows the instruction that was executed immediately before the interrupt.

Note:

The interrupt request flag bit of the peripheral function is not automatically cleared when the interrupt request is accepted. The interrupt processing routine must be used to clear the interrupt request by programming (usually, writing 0 to the interrupt request flag bit).

Standby mode (low power consumption) is released by an interrupt. See Section 3.7 "Standby Mode (Low Power Consumption)", for details.

Reference:

If the interrupt request flag bit is cleared at the beginning of the interrupt processing routine, the peripheral function that caused the interrupt can cause another interrupt (reset the interrupt request flag bit) during execution of the interrupt processing routine. However, the interrupt is usually accepted after the current interrupt processing routine has terminated.

3.4.3 Multiple Interrupts

Multiple interrupts can be generated for multiple interrupt requests from peripheral functions by setting different interrupt levels in the interrupt level registers (ILR1, ILR2, and ILR3).

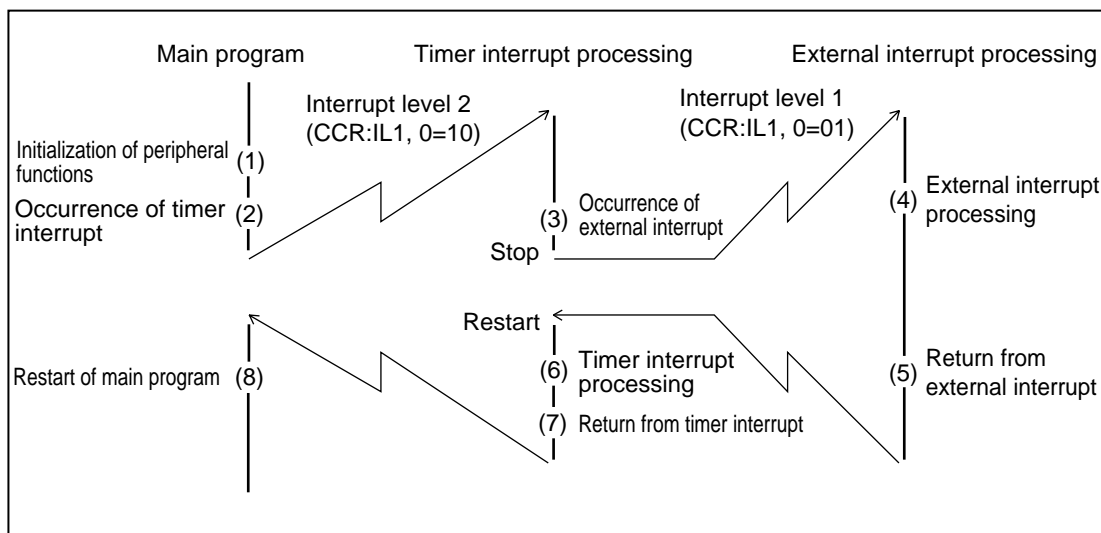
■ Multiple Interrupts

If an interrupt request set at higher interrupt level occurs during execution of the interrupt processing routine, the higher interrupt request is accepted. Interrupt levels of 1 to 3 can be set. If level 3 is set, however, the CPU does not accept any interrupt requests.

■ Example of Multiple Interrupts

The example of multiple interrupt processing assumes that an external interrupt has precedence over a timer interrupt. The timer interrupt level is set to 2, and the external interrupt level is set to 1. If an external interrupt occurs during timer interrupt processing, the processing shown in Figure 3.4-3 "Example of Multiple Interrupts" is executed.

Figure 3.4-3 Example of Multiple Interrupts



- During timer interrupt processing, the interrupt level bits (CCR: IL1 and IL0) of the condition code register have the same value (2 in the example) as the value of the interrupt level register (ILR1, ILR2, or ILR3) corresponding to the timer interrupt. If an interrupt request set at a higher interrupt level (1 in the example) occurs, that interrupt is processed first.
- To temporarily disable multiple interrupts during a timer interrupt, set the interrupt enable flag in the condition code register to interrupt disabled (CCR: I = 0) or the interrupt level bits (IL1 and IL0) to 00.
- When interrupt processing terminates and an interrupt return instruction (RETI) is executed, the program counter (PC) and program status (PS) values saved on the stack are restored to return to processing of the interrupted program. The condition code register (CCR) has the value it had before the interrupt by restoring the program status (PS) value.

3.4.4 Interrupt Processing Time

The time required until control is transferred to the interrupt processing routine after an interrupt request occurs is the sum of the time required for termination of the instruction currently being executed and the interrupt handling time (time required to prepare interrupt processing). This time can be up to 30 instruction cycles.

■ Interrupt Processing Time

An interrupt request sample wait time and interrupt handling time are required from the occurrence of an interrupt until acceptance of the interrupt and execution of the interrupt processing routine.

○ Interrupt request sample wait time

An interrupt request is sampled in the last cycle of each instruction to determine whether the interrupt request has occurred. During execution of an instruction, the CPU is unable to recognize an interrupt request. The wait time becomes the maximum wait time if an interrupt request occurs immediately after execution of the DIVU instruction (at 21 instruction cycles, the longest execution cycle) is started.

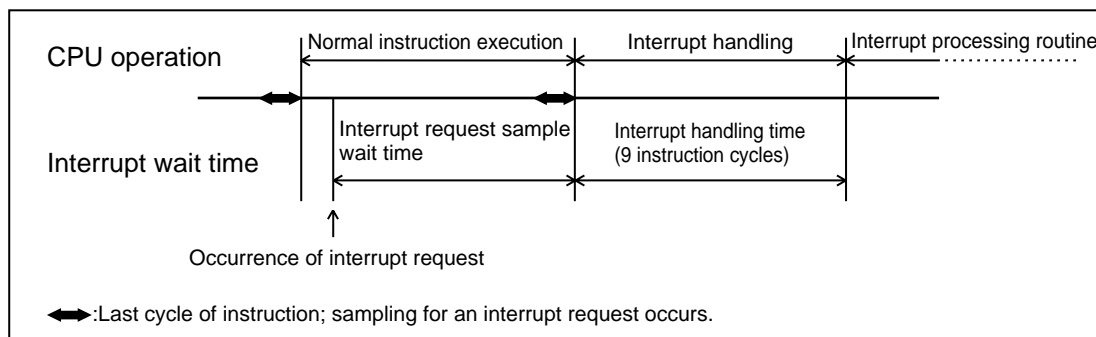
○ Interrupt handling time

The CPU requires 9 instruction cycles to accept an interrupt and to prepare interrupt processing, as follows:

- Saving the program counter (PC) and program status (PS)
- Setting the first address (interrupt vector) of the interrupt processing routine in the PC
- Updating the interrupt level bit (PS: CCR: IL1 and IL0) in the program status (PS)

Figure 3.4-4 "Interrupt Processing Time" shows the interrupt processing time.

Figure 3.4-4 Interrupt Processing Time



If an interrupt request occurs immediately after execution of the DIVU instruction (at 21 instruction cycles, the longest execution cycle) is started, an interrupt processing time of $21 + 9 = 30$ instruction cycles is required. However, if the DIVU and MULU instructions are not used in the program, an interrupt processing time of $6 + 9 = 15$ instruction cycles is required.

3.4.5 Stack Operation During Interrupt Processing

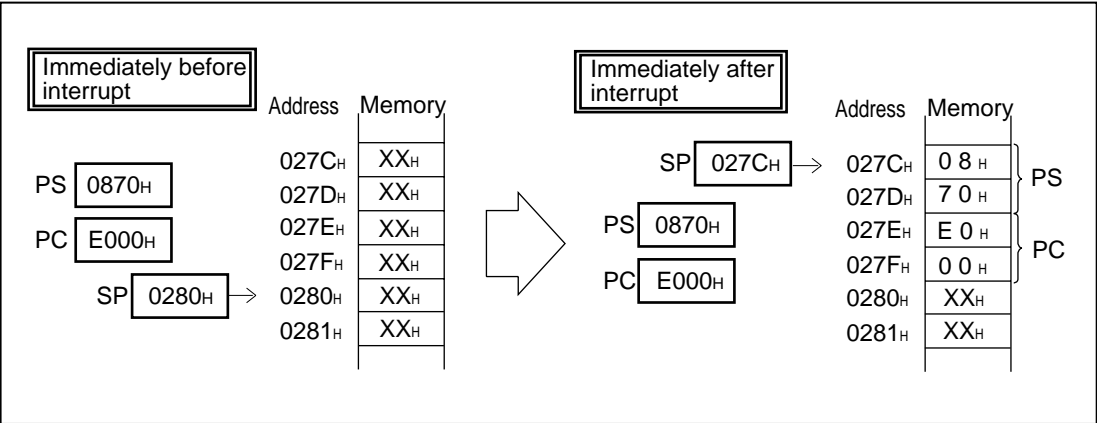
This section explains how registers are saved and restored during interrupt processing.

■ Stack Operation When Interrupt Processing Starts

When an interrupt is accepted, the CPU automatically saves the contents of the current program counter (PC) and the program status (PS) on the stack.

Figure 3.4-5 "Stack Operation When Interrupt Processing Starts" shows stack operation when interrupt processing starts.

Figure 3.4-5 Stack Operation When Interrupt Processing Starts



■ Stack Operation on Return from an Interrupt

When an interrupt return instruction (RETI) is executed at the termination of interrupt processing, the program status (PS) and program counter (PC) values are restored from the stack in this order, which is the reverse of the order when interrupt processing starts. This processing restores their PS and PC states at the point immediately before the interrupt started.

Note:

The accumulator (A) and temporary accumulator (T) are not automatically saved on the stack. Use the PUSHW and POPW instructions to save and restore the A and T values.

3.4.6 Stack Area for Interrupt Processing

The stack area in RAM is used to execute interrupt processing. The contents of the stack pointer (SP) indicate the first address of the stack area.

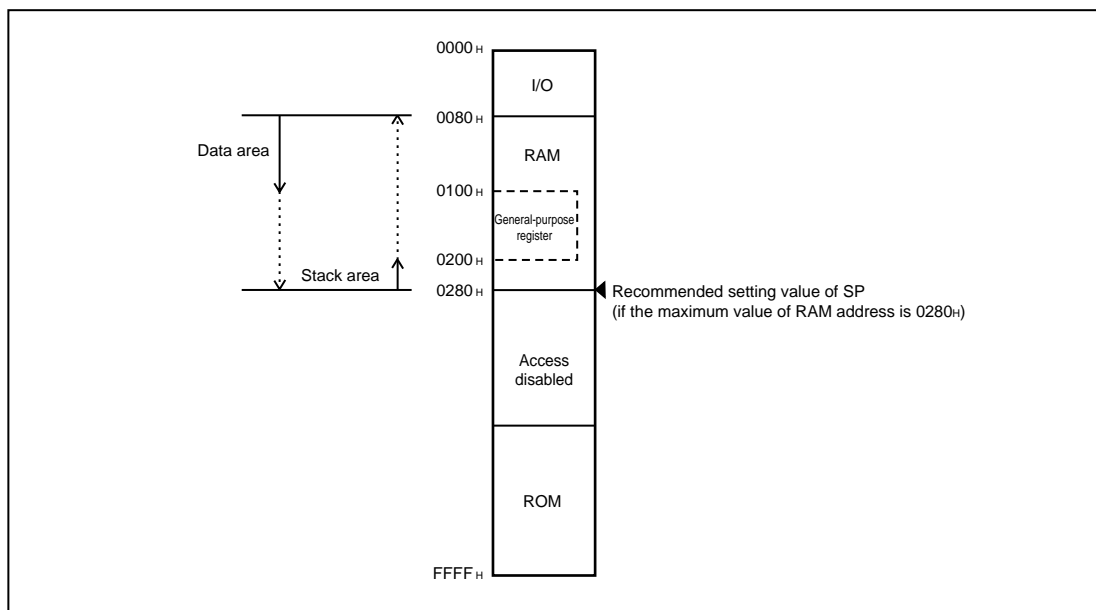
■ Stack Area for Interrupt Processing

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

- The stack area is allocated in RAM with the data area.
- The stack pointer (SP) should be initialized to indicate the maximum RAM address value. The data area should be provided at a lower RAM address.

Figure 3.4-6 "Stack Area for Interrupt Processing" shows an example of setting the stack area.

Figure 3.4-6 Stack Area for Interrupt Processing



Reference:

Interrupts, subroutine calls, and the PUSHW instruction use the stack area in descending order of address value. The return instructions (RETI and RET) and POPW instruction release the stack area in ascending order of address value. Be sure that the stack area used by multiple interrupts and subroutine calls does not overlap the data area and general-purpose register area used to hold other data when the address value of the stack area becomes smaller.

3.5 Resets

A reset results from one of the following four reset causes:

- External reset
- Software reset
- Watchdog reset
- Power-on reset

■ Reset Causes

Table 3.5-1 Reset Causes

Reset cause	Reset condition
External reset	The external reset pin is set to the L level.
Software reset	0 is written to the software reset bit (STBC: RST) of the standby control register.
Watchdog reset	Watchdog timer overflows.
Power-on reset	Power is turned on.

○ External reset

An external reset causes a reset by inputting the L level to the external reset pin (RST). When the reset pin is set to H level, the external reset is released.

The external reset pin also functions as the reset output pin.

○ Software reset

A software reset causes a reset of 4 instruction cycles by writing 0 to the software reset bit (STBC: RST) of the standby control register.

○ Watchdog reset

A watchdog reset causes a reset of 4 instruction cycles when data is not written to the watchdog control register (WDTC) within the specified time after activation of watchdog timer.

○ Power-on reset

A power-on reset causes a reset by turning on the power.

■ Reset Causes and Main Clock Oscillation Stabilization Wait Time

The oscillation stabilization wait time depends on the operating mode when a reset occurs.

After termination of a reset, normal operation is started in main clock mode regardless of operating mode (clock mode and standby mode) before the reset and the reset cause. If a reset occurs when main clock oscillation stops or during the main clock oscillation stabilization wait time, the main clock oscillation stabilization wait reset state is entered.

For a software reset and watchdog reset, during operation in main clock mode, an oscillation stabilization wait time is not required.

Table 3.5-2 "Reset Causes and Oscillation Stabilization Wait Time" lists the relationship between reset causes and main clock oscillation stabilization wait time and the reset operation (mode fetch).

Table 3.5-2 Reset Causes and Oscillation Stabilization Wait Time

Reset cause	Operation state	Reset operation and main clock oscillation stabilization wait time
External reset ^{*1}	At power-on and in stop mode	If an external reset is released after the main clock oscillation stabilization wait time elapses, the reset operation is performed ^{*2} .
Software reset and watchdog reset	Main clock mode	After a reset of 4 instruction cycles occurs, the reset operation is performed ^{*3} .
Power-on reset		After power-on, the main clock oscillation stabilization wait time elapses, and then the reset operation is performed ^{*2} .

*1: During operation in main clock mode, an external reset does not require the oscillation stabilization wait time. After the external reset is released, the reset operation is performed.

*2: During the main clock oscillation stabilization wait time, the L level is output to the $\overline{\text{RST}}$ pin.

*3: Within 4 instruction cycles, the L level is output to the $\overline{\text{RST}}$ pin.

3.5.1 External Reset Pin

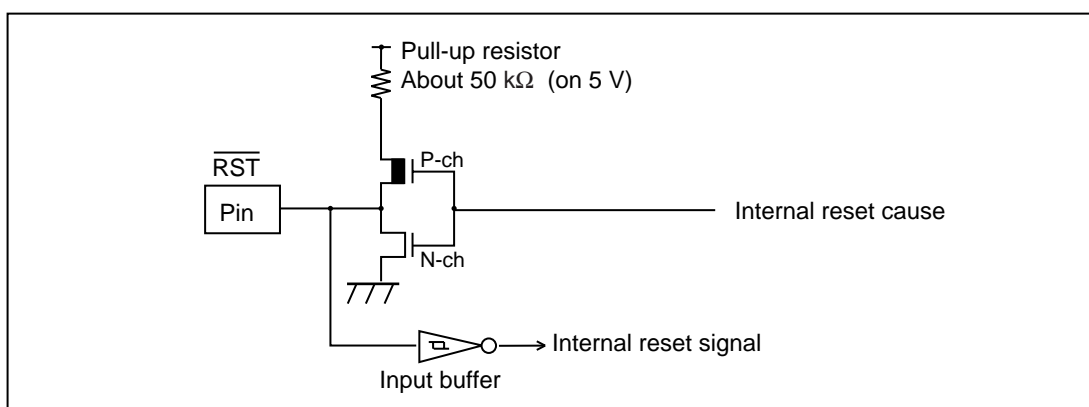
An external reset pin causes a reset when the L level is input to it. The external reset pin outputs the L level for an internal reset cause.

■ Block Diagram of the External Reset Pin

The external reset pin ($\overline{\text{RST}}$) of a model is set to hysteresis input and pulled up N-ch open drain output.

Figure 3.5-1 "Block Diagram of External Reset Pin" is a block diagram of the external reset pin.

Figure 3.5-1 Block Diagram of External Reset Pin



■ Functions of the External Reset Pin

The external reset pin ($\overline{\text{RST}}$) generates an internal reset signal by inputting the L level.

The external reset pin outputs the L level for an internal reset cause and based on the oscillation stabilization wait time caused by a reset. Internal reset causes include software resets, watchdog resets, and power-on resets.

Note:

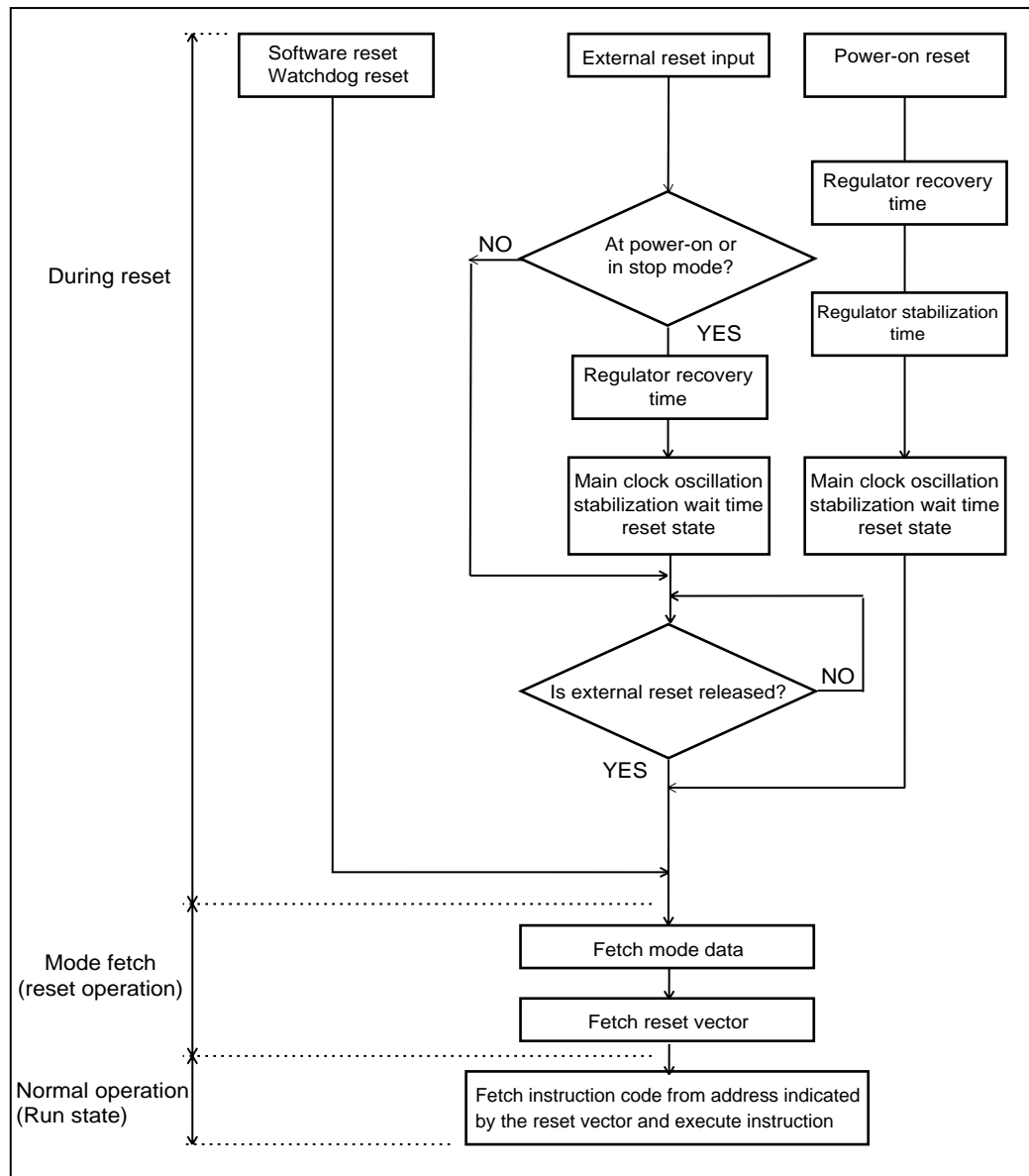
External reset input is accepted asynchronously regardless of the internal clock. Initialization of the internal circuit requires a clock.

3.5.2 Reset Operation

When a reset is released, the CPU reads mode data and the reset vector from the internal ROM according to the setting of the mode pin (mode fetch). At power-on, to return from stop mode after a reset, mode fetch is performed after the oscillation stabilization wait time elapses. If a reset occurs, the RAM contents are not assured.

■ Overview of Reset Operation

Figure 3.5-2 Flow of Reset Operation



CHAPTER 3 CENTRAL PROCESSING UNIT (CPU)

■ Mode Pin

The MB89580B/BW is dedicated to single-chip mode. Be sure to set the mode pins (MOD0 and MOD1) to Vss. Select the internal ROM from which mode data and the reset vector will be read. Do not change the setting of the mode pin after termination of the reset operation.

■ Mode Fetch

When a reset is released, mode data and the reset vector are read from the internal ROM.

○ Mode data (address: FFFD_H)

Be sure to set single-chip mode (00_H) for the mode data.

○ Reset Vector (address: Higher FFFE_H/lower FFFF_H)

After termination of the reset operation, the address at which execution started is written. An instruction is executed at the address contained in the reset vector.

■ Oscillation Stabilization Wait Time Reset State

Reset operations for a power-on reset and an external reset in stop mode are performed after the oscillation stabilization wait time elapses. Unless external reset input is released, the reset operation is performed after the external reset is released.

Because the oscillation stabilization wait time is required, an external clock must be input for a reset.

The main clock oscillation stabilization wait time is generated by the timebase timer.

■ Effect of a Reset on the Contents of RAM

An external reset is not synchronized with the internal clock but provides an internal reset directly. The contents of RAM may be changed before and after the reset. Be sure to initialize RAM before it is used.

■ Regulatory Recovery Time

The MB89580B/BW incorporates a voltage step-down circuit to ensure operation at 3.3 V. The voltage step-down circuit reduces power consumption to a minimum level during stop mode. The regulator recovery time, which is the time required for the voltage reduction circuit to restart normal operation, requires 20 μs or longer.

■ Regulatory Stabilization Time

Power-on reset requires a regulatory stabilization time in addition to an oscillation stabilization waiting time. The voltage supplied to the power source terminals is required to reach the minimum operating voltage within the regulatory stabilization time.

The regulatory stabilization time requires $2^{19}/F_{ch}$ or more, depending on the oscillation frequency (F_{ch} : main clock oscillation frequency).

3.5.3 State of Pins After a Reset

A reset initializes pins.

■ Pin State During Reset

If a reset cause occurs, all pins (resource pins) except some I/O pins enter the high impedance state, and mode data is read from the internal ROM.

■ Pin State After Mode Data Is Read

Most I/O pins remain in the high impedance state immediately after mode data is read (pins with pull-up resistor selected in the pull-up option register are set to the H level).

Note:

Be sure that the equipment connected to a pin that enters the high impedance state when a reset cause occurs does not cause a malfunction.

3.6 Clock

A clock is generated by connecting a resonator externally. A clock is input to the clock controller, then it is multiplied by the PLL circuit and divided as the CPU operating clock and peripheral circuit operating clock. Supply of these operating clocks is controlled according to the operating mode of low power consumption mode.

■ Clock Supply Map

Oscillation of a clock and supply of a clock to the peripheral circuit (peripheral functions) are controlled by the clock controller. The operating clocks of the CPU and peripheral circuits are affected by standby mode (sleep or stop).

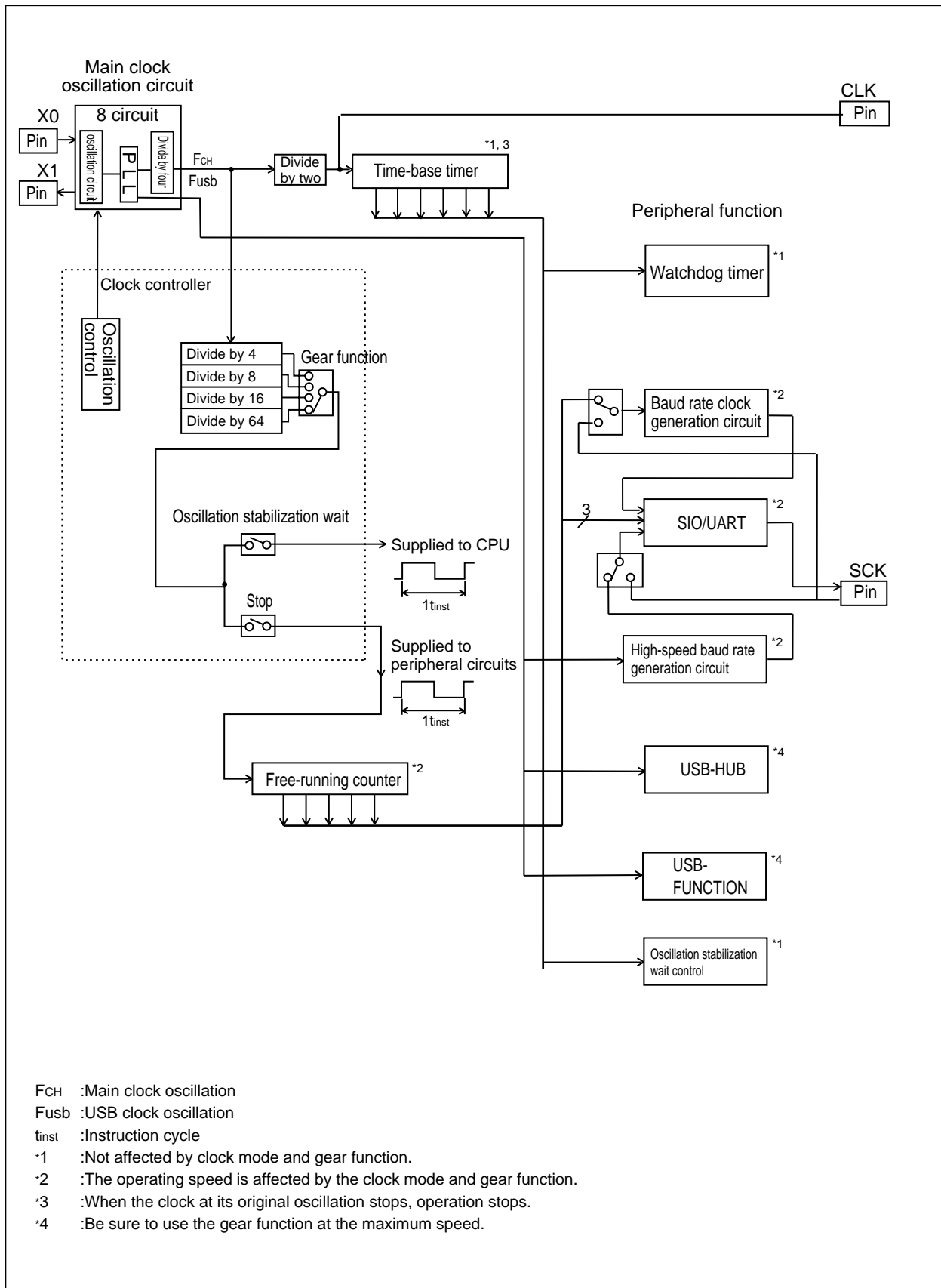
Divide-by output of the free-running counter operating on the clock for the peripheral circuits is supplied to the peripheral functions. However, peripheral functions to which HCLK output that provides divide-by-two output of the main clock oscillation and divide-by output of the timebase timer are supplied are not affected by the gear function.

The USB function section (in full speed operation mode) operates on a clock multiplied by 8 by the PLL circuit (48 MHz) and the divide-by-four clock (12 MHz).

In low speed operation mode, the USB function section operates at 6 MHz and on the divide-by-four clock (1.5 MHz).

Figure 3.6-1 "Clock Supply Map" shows the clock supply map.

Figure 3.6-1 Clock Supply Map



3.6.1 Clock Generator

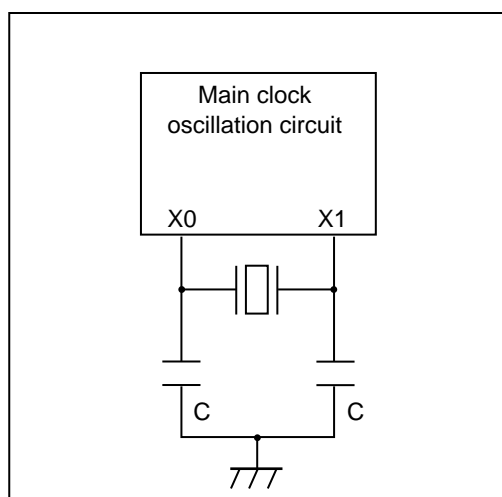
Stop mode controls enabling and stopping of the main clock oscillation.

■ Clock Generator

○ For crystal resonator

Connect as shown in Figure 3.6-2 "Example of Connecting a Crystal Resonator".

Figure 3.6-2 Example of Connecting a Crystal Resonator



3.6.2 Clock Controller

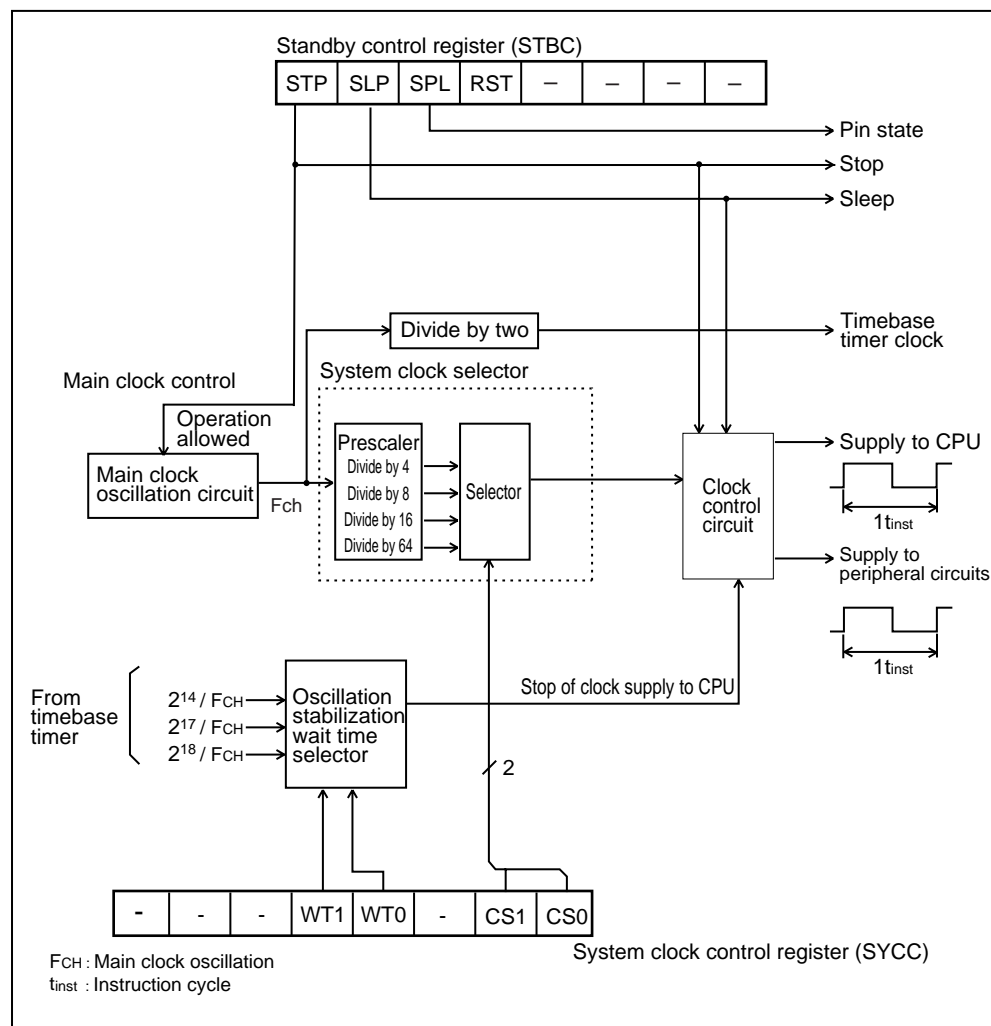
The clock controller consists of the following six blocks:

- Main clock oscillation circuit
- System clock selector
- Clock control circuit
- Oscillation stabilization wait time selector
- System clock control register (SYCC)
- Standby control register (STBC)

■ Block Diagram of the Clock Controller

Figure 3.6-3 "Block Diagram of Clock Controller" is a block diagram of the clock controller.

Figure 3.6-3 Block Diagram of Clock Controller



CHAPTER 3 CENTRAL PROCESSING UNIT (CPU)

○ **Main clock oscillation circuit**

This circuit oscillates the main clock and stops oscillation in stop mode.

○ **Clock control circuit**

This circuit controls supply of the operating cycle to the CPU and peripheral circuits based on the standby mode (sleep or stop).

This circuit stops supply of the clock to the CPU until the clock supply stop signal of the oscillation stabilization wait time selector is released.

○ **Oscillation stabilization wait time selector**

This selector selects one of three main clock oscillation stabilization wait times generated by the timebase timer (determined by the standby mode or reset), and outputs it as a clock supply stop signal to the CPU.

○ **System clock control register (SYCC)**

This register selects the main clock speed and main clock oscillation stabilization wait time.

○ **Standby control register (STBC)**

This register switches from normal operation (run) to standby mode, sets the state of pins in stop mode, and performs a software reset.

○ **System clock selector**

This selector selects one of four clocks that are divisions of the main clock oscillation and supplies it to the clock control circuit.

3.6.3 System Clock Control Register (SYCC)

The system clock control register (SYCC) selects the main clock speed and oscillation stabilization wait time.

■ System Clock Control Register (SYCC)

Figure 3.6-4 Configuration of System Clock Control Register (SYCC)

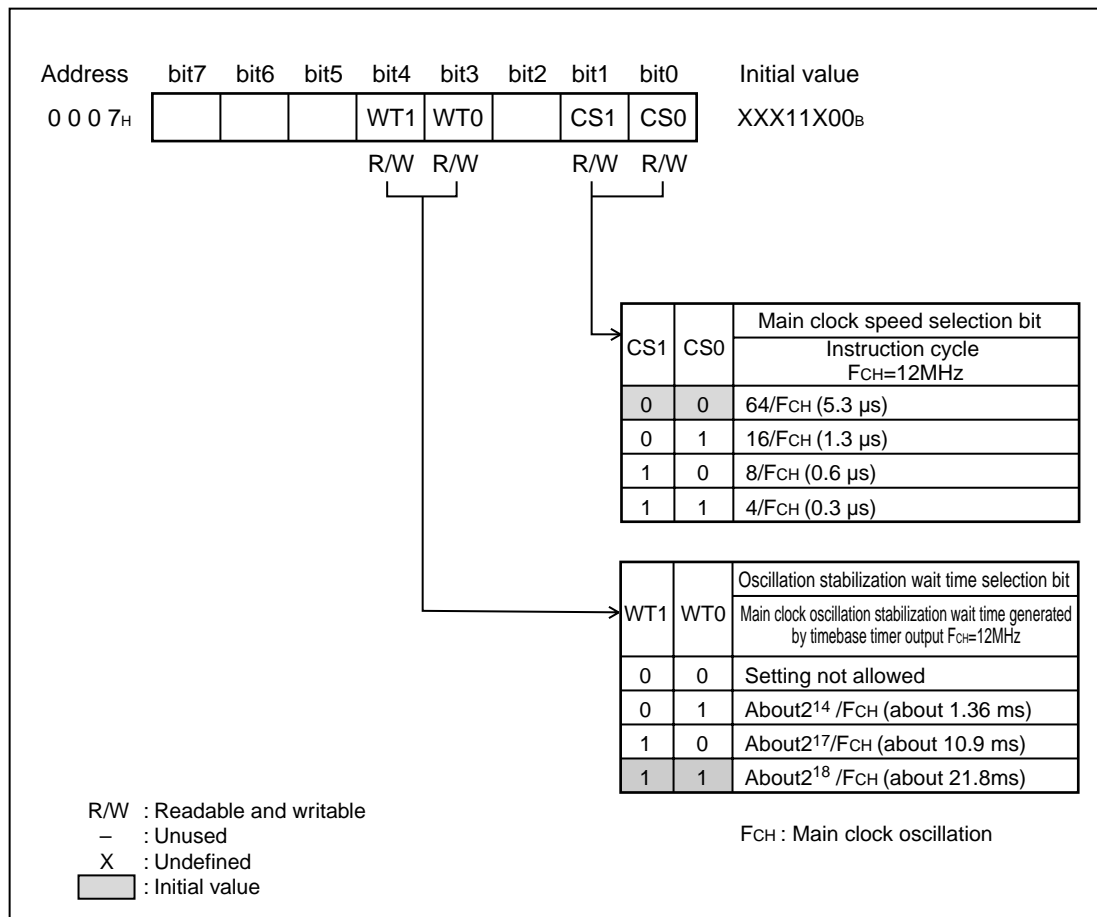


Table 3.6-1 Functions of System Clock Control Register (SYCC) Bits

Bit name		Function
bit7 bit6 bit5	Unused	<ul style="list-style-type: none"> Read values are undefined. Writing has no effect on operation.
bit4 bit3	WT1,WT0: Oscillation stabilization wait time selection bit	<ul style="list-style-type: none"> This bit selects the main clock oscillation stabilization wait time. When an external interrupt returns control from stop mode to normal operation, the oscillation stabilization wait time selected by this bit is used. The initial values of these bits are WT1 = 1 and WT0 = 1.
bit2	Unused	Note: Always write 1.
bit1 bit0	CS1,CS0: Main clock speed selection bit	<ul style="list-style-type: none"> This bit selects the clock speed. Four speeds for operating clocks for the CPU and peripheral functions can be set (gear function). However, these bits do not affect the operating clock for the timebase timer.

■ Instruction Cycle (t_{inst})

The instruction cycle (minimum execution time) can be selected from divide-by four, divide-by eight, divide-by-16, and divide-by-64 clocks of the main clock using the main clock selection bits (CS1 and CS0) of the SYCC register.

In main clock mode (SYCC: CS1, SC0 = 11_B), the instruction cycle becomes $4/F_{CH}$ = about 0.33 μ s if main clock oscillation (F_{CH}) is 12 MHz.

Before enabling the USB circuit function, be sure to set this instruction cycle to divide-by-four.

3.6.4 Clock Mode

The main clock speed is switched by selection from among four clocks generated by dividing the main clock oscillation.

■ Clock Mode Operating State

Table 3.6-2 Clock Mode Operating State

Clock mode	Main clock speed SYCC register (CS1,CS0)		Standby mode	Clock	Operating clock for sections			Cause of release of standby mode (except reset)
					CPU	Time base timer	Peripheral functions	
Main clock mode	(1.1)	<div>High speed</div> <div>↑</div> <div>↓</div> <div>Low speed</div>	RUN	Oscillation	F _{CH} /4	F _{CH} /2	F _{CH} /4	Interrupt request
			Sleep					
			Stop	Stop	Stop	Stop	Stop	External interrupt and USB interrupt
	(1.0)		RUN	Oscillation	F _{CH} /8	F _{CH} /2	F _{CH} /8	Interrupt request
			Sleep					
			Stop	Stop	Stop	Stop	Stop	External interrupt and USB interrupt
	(0.1)		RUN	Oscillation	F _{CH} /16	F _{CH} /2	F _{CH} /16	Interrupt request
			Sleep					
			Stop	Stop	Stop	Stop	Stop	External interrupt and USB interrupt
	(0.0)		RUN	Oscillation	F _{CH} /64	F _{CH} /2	F _{CH} /64	Interrupt request
			Sleep					
			Stop	Stop	Stop	Stop	Stop	External interrupt and USB interrupt

F_{CH} : Main clock oscillation

The corresponding standby mode can be entered from each clock mode. See Section 3.7 "Standby Mode (Low Power Consumption)", for information about standby mode.

■ Operation in Main Clock Mode

If standby mode is specified, sleep mode or stop mode can be entered.

Regardless of which type of reset occurs, operation is always started in main clock run mode (operating modes are released by a reset).

3.6.5 Oscillation Stabilization Wait Time

At power-on or in main stop mode, when the main clock stops, the main clock stabilization wait time is required for operation in main clock run mode.

■ Oscillation Stabilization Wait Time

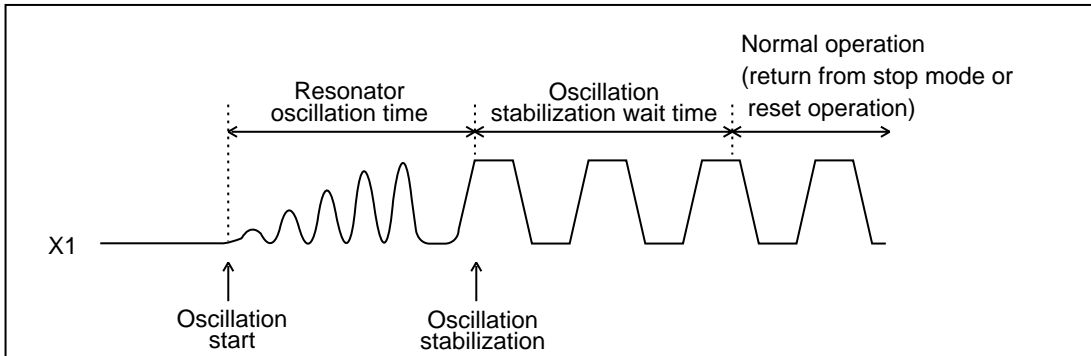
For stabilization at a proper oscillation (oscillation frequency), the crystal resonator generally requires from a few milliseconds to some tens of milliseconds between the start of oscillation and its stabilization.

Immediately after oscillation starts, CPU operation is disabled. When the oscillation stabilization wait time elapses and oscillation is completely stabilized, a clock is supplied to the CPU.

Since the oscillation stabilization time depends on the type of the resonator connected to the oscillator (clock generator), the oscillation stabilization wait time appropriate to the resonator being used must be selected.

Figure 3.6-5 "Oscillator Operation Immediately After Oscillation Starts" shows oscillator operation immediately after oscillation starts.

Figure 3.6-5 Oscillator Operation Immediately After Oscillation Starts



■ Main Clock Oscillation Stabilization Wait Time

When main clock oscillation stops, a main clock oscillation stabilization wait time is required before operation starts.

Incrementing of the main clock oscillation stabilization wait time starts when the timebase timer counter is cleared, and ends when the specified bit overflows.

○ Oscillation stabilization wait time during operation

The oscillation stabilization wait time selection bits (SYCC: WT1, WT0) of system control register can be used to select one of three oscillation stabilization wait times for returning from stop mode to main clock run mode in response to an external interrupt.

○ Oscillation stabilization wait time for a reset

The oscillation stabilization wait time (initial values of WT1, WT0) for a reset is a fixed value.

Table 3.6-3 Conditions for Starting Main Clock Mode Operation and Oscillation Stabilization Wait Time

Main clock mode operation start condition	At power-on	Stop mode release		
		External reset	External interrupt	USB interrupt
Oscillation stabilization wait time selection	SYCC:WT1=1,WT0=1 *	SYCC:WT1,WT0 *		

*: Oscillation stabilization wait time selection bit of the system clock control register

3.7 Standby Mode (Low Power Consumption)

Standby mode consists of a sleep mode and stop mode.

Entry into standby mode is controlled by the setting of the standby control register (STBC).

In main clock mode, two modes, sleep and stop, can be entered.

Stopping the operation of the CPU and peripheral functions in standby mode reduces power consumption.

This section explains the relationship between standby mode and clock mode and the operating state of each section in standby mode.

■ Standby Mode

In standby mode, power consumption is reduced by using the clock controller to stop supply of the clock to the CPU (sleep mode) and stopping oscillation (stop mode).

○ Sleep mode

Sleep mode stops operation of the CPU and watchdog timer, and operates on the main clock.

○ Stop mode

Stop mode stops operation of the CPU and peripheral functions, and stops all functions except external and USB interrupts.

3.7.1 Standby Mode Operating State

This section explains operating states of the CPU and peripheral functions in standby mode.

■ Standby Mode Operating State

Table 3.7-1 Operating States of CPU and Peripheral Functions in Standby Mode

Operating mode		Main clock mode			
Function		RUN	Sleep	Stop (SPL=0)	Stop (SPL=1)
Main clock		Operation	Operation	Stop	Stop
CPU	Instruction	Operation	Stop	Stop	Stop
	ROM	Operation	Hold	Hold	Hold
	RAM				
Peripheral function	I/O port	Operation	Hold	Hold	Hold
	Timebase timer	Operation	Operation	Stop	Stop
	Watchdog timer	Operation	Stop	Stop	Stop
	8-bit PWM timer 1, 2	Operation	Operation	Stop	Stop
	8-bit serial I/O	Operation	Operation	Stop	Stop
	External interrupt	Operation	Operation	Operation	Operation
	USB function	Operation	Operation	Stop	Stop
Pin		Operation	Hold	Hold	Hi-Z
Release method		Reset and interrupts	Reset and interrupts	Reset and interrupts	

○ State of pins in standby mode

The pin state selection bit (STBC: SPL) of the standby control register can be used to keep most I/O pins in the state there were in immediately before entry to stop mode or in high impedance state without relation to clock mode.

3.7.2 Sleep Mode

This section explains operation in sleep mode.

■ Operation in Sleep Mode

○ Entering sleep mode

Sleep mode stops the operating clock of the CPU. The CPU retains the contents of registers and RAM existing immediately before sleep mode is entered and stops. The peripheral functions except the watchdog timer continue to operate.

Sleep mode is entered by writing 1 to the sleep bit (STBC: SLP) of the standby control register. If an interrupt request occurs when 1 is written to the SLP bit, the write is ignored and execution of instruction continues without sleep mode being entered (sleep mode is not entered after interrupt processing).

○ Releasing sleep mode

Sleep mode is released by a reset and an interrupt from the peripheral functions.

Reset in main sleep mode does not require an oscillation stabilization wait time.

The states of pins are initialized by a reset operation.

If peripheral functions or an external interrupt circuit issues an interrupt request in sleep mode at a request level that is higher than 11, sleep mode is released regardless of the setting of CPU interrupt enable bit (CCR: I) and interrupt level bits (CCR: IL1, IL0).

After release, normal interrupt operation is performed. If an interrupt is accepted, interrupt processing is executed. If an interrupt is not accepted, processing is executed starting with the instruction following the instruction that was executed immediately before sleep mode was entered.

3.7.3 Stop Mode

This section explains operation in stop mode.

■ Operation in Stop Mode

○ Entering stop mode

Stop mode stops oscillation. The contents of registers and RAM immediately before stop mode was entered are retained, and most functions stop.

In main clock mode, main clock oscillation stops. Peripheral functions and the CPU stop operation except the external interrupt circuit and USB interrupt circuit. Data is preserved using the least amount of power.

Stop mode is entered by writing 1 to the stop bit (STBC: STP) of the standby control register. If the pin state selection bit (STBC: SPL) is 0, the state of the power-on pin is retained. If the bit is 1, the external pin is placed in the high-impedance state (the external pin with pull-up resistor selected in the pull-up setting register is set to the H level).

If an interrupt request occurs when 1 is written to the STP bit, the write is ignored and execution of the instruction continues without stop mode being entered (stop mode is not entered after interrupt processing).

To enter stop mode, disable interrupt request output of the timebase timer (TBTC: TBIE = 0) if this is required.

○ Releasing stop mode

Stop mode is released by a reset or an external interrupt or USB interrupt.

If a reset occurs in stop mode, the reset operation is performed after the main clock oscillation stabilization wait time elapses.

The states of pins are initialized by a reset.

If the external interrupt circuit issues an interrupt request during stop mode at an interrupt level that is higher than 11, stop mode is released regardless of the setting of the CPU interrupt enable bit (CCR: I) and interrupt level bits (CCR: IL1, IL0). In stop mode, peripheral functions stop, and no interrupt requests, except external and USB interrupts, are not issued.

After release, normal interrupt operation is performed after the oscillation stabilization wait time elapses. If an interrupt is accepted, interrupt processing is executed. If an interrupt is not accepted, processing is executed starting with the instruction following the instruction that was executed immediately before stop mode was entered.

When stop mode is released by an external interrupt, some peripheral functions restart operations from where they stopped. The initial interval time of the interval timer function is undefined. Be sure that peripheral functions are initialized after a return from stop mode.

Reference:

Stop mode is only released by interrupt requests from the external interrupt circuit and USB interrupt circuit.

3.7.4 Standby Control Register (STBC)

The standby control register (STBC) is used to switch to sleep mode or stop mode, to set pin states in stop mode, and to perform a software reset.

■ Standby Control Register (STBC)

Figure 3.7-1 Standby Control Register (STBC)

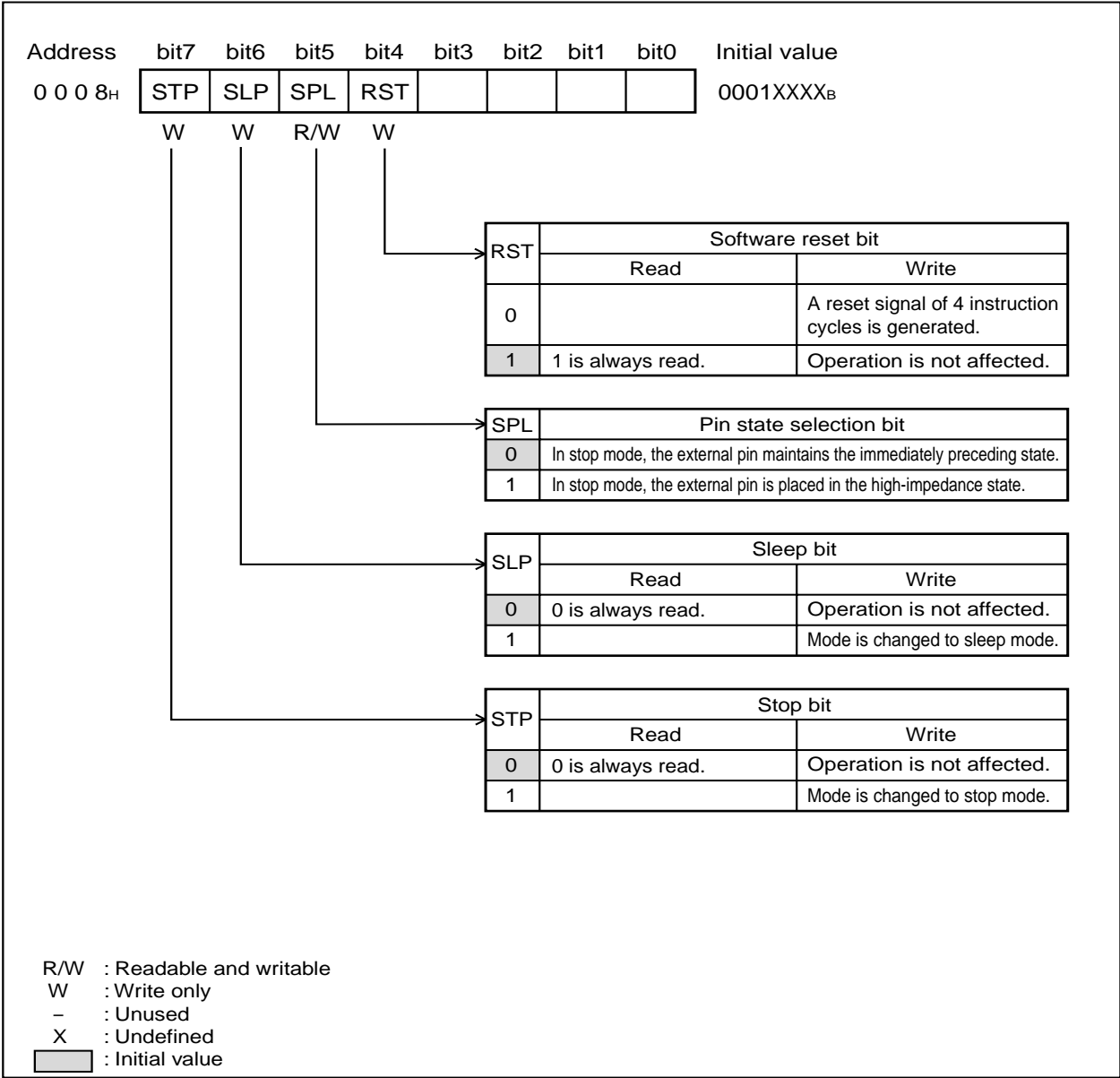


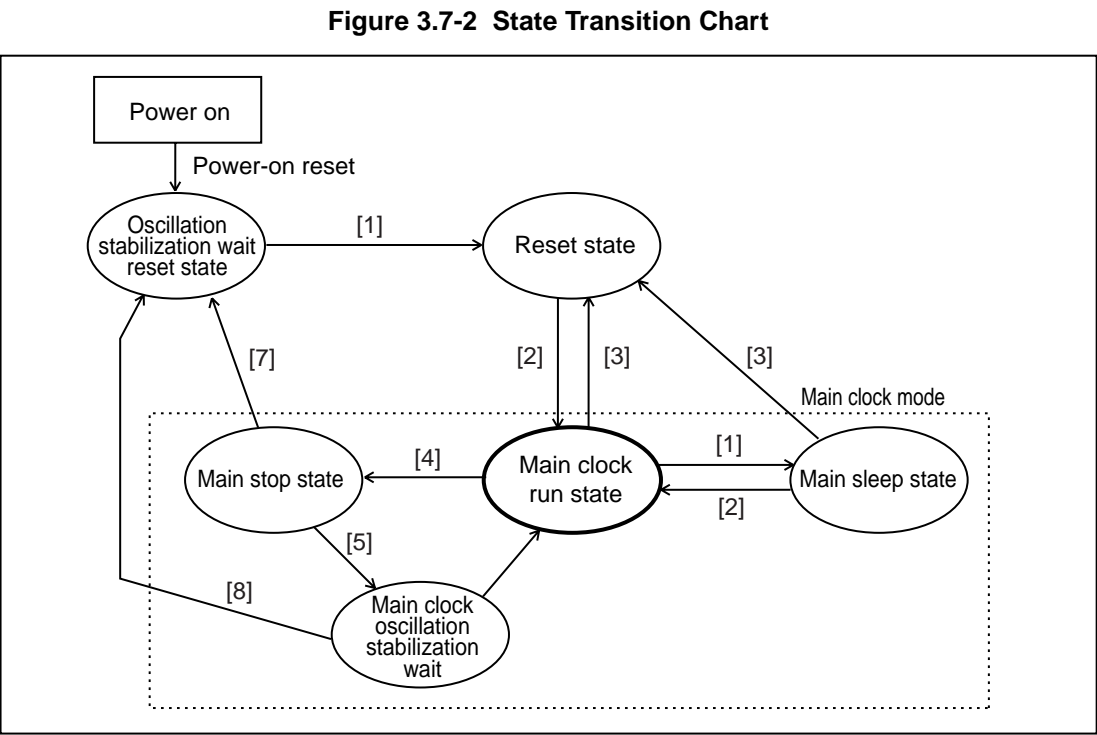
Table 3.7-2 Functions of Standby Control Register (STBC) Bits

Bit name		Function
bit7	STP: Stop bit	<ul style="list-style-type: none"> • This bit specifies that stop mode is entered. • Stop mode is entered by writing 1 to this bit. • Writing 0 has no effect on operation. • When this bit is read, 0 is always read.
bit6	SLP: Sleep bit	<ul style="list-style-type: none"> • This bit specifies that stop mode is entered. • Stop mode is entered by writing 1 to this bit. • Writing 0 has no effect on operation. • When this bit is read, 0 is always read.
bit5	SPL: Pin state selection bit	<ul style="list-style-type: none"> • This bit selects the state of the power-on pin in stop mode. • If 0 is written to this bit, the state (level) of the power-on pin is retained when stop mode is entered. • If 1 is written, the external pin enters the high-impedance state when stop or clock mode is entered (the external pin with pull-up resistor selected in the pull-up setting register is set to the H level). • A reset sets this bit to 0.
bit4	RST: Software reset bit	<ul style="list-style-type: none"> • This bit selects a software reset. • An internal reset cause of 4 instruction cycles is generated by writing 0 to this bit. • Writing 1 has no effect on operation. • If this bit is read, 1 is always read.
bit3 bit2 bit1 bit0	Unused	<ul style="list-style-type: none"> • Always write 1. • Read values are undefined. • Writing has no effect on operation.

3.7.5 State Transition Chart

Figure 3.7-2 "State Transition Chart" shows the state transition chart.

■ State Transition Chart



○ Entering normal state (RUN) and reset

Table 3.7-3 Entering Main Clock Run Mode and Reset

State transition	Transition condition (Figure 3.7-2 "State Transition Chart")
After power-on, entering normal state (RUN)	[1] Main clock oscillation stabilization wait time ends (timebase timer output)
	[2] Reset input is released
Reset in the run state	[3] External reset, software reset, and watchdog reset

○ Entering and releasing standby mode

Table 3.7-4 Entering and Releasing Standby Mode

State transition	Transition condition (Figure 3.7-2 "State Transition Chart")
Entering sleep mode	[1] STBC:SLP=1
Releasing sleep mode	[2] Interrupts (all types) [3] External reset
Entering stop mode	[4] STBC:STP=1
Releasing stop mode	[5] External interrupt or USB interrupt [6] Main clock oscillation stabilization wait time ends (timebase timer output) [7] External reset [8] External reset (during oscillation stabilization wait)

STBC: Standby control register

3.7.6 Notes on Using Standby Mode

Even though standby mode is set in the standby control register (STBC), standby mode is not entered if an interrupt request is issued from a peripheral function. If an interrupt returns from standby mode to the normal operating state, operation after the return depends on whether the interrupt request is accepted.

■ Entering Standby Mode and Interrupts

An interrupt request with an interrupt priority that is higher than 11 may be issued from a peripheral function to the CPU. In this case, if 1 is written to the stop bit (STBC: STP) and the sleep bit (SLP) of the standby control register, the write is ignored. Standby mode is not entered (standby mode is not entered after interrupt processing).

This has no relation to whether or not the CPU accepts an interrupt.

Even though the CPU is processing an interrupt, the interrupt request flag bit is cleared. Unless there is another interrupt request, standby mode can be entered.

■ Releasing Standby Mode with an Interrupt

When an interrupt request with interrupt priority that is higher than 11 is issued from a peripheral function in sleep or stop mode, standby mode is released. This has no relation to whether or not the CPU accepts an interrupt.

After release, the priority of the interrupt level register (ILR1 to ILR3) corresponding to the interrupt request may be higher than the interrupt level bits (CCR: IL1 and IL0) of the condition code register and the interrupt enable flag may be set to enable (CCR: I = 1). In this case, as part of normal interrupt operation, a branch to the interrupt processing routine occurs. If the interrupt is not accepted, operation is restarted with the instruction following the instruction that started standby mode.

If a branch to the interrupt processing routine is not made immediately after return, action must be taken to disable interrupts before the standby mode is set.

■ Note on Setting Standby Mode

Set standby mode in the standby control register (STBC) according to Table 3.7-5 "Setting of Low Power Consumption Mode in the Standby Control Register (STBC)". If 1 is written to these bits concurrently, stop mode has precedence over sleep mode. Be sure that 1 is set to only one of the bits.

Table 3.7-5 Setting of Low Power Consumption Mode in the Standby Control Register (STBC)

STBC register		Mode
STP(bit7)	SLP(bit6)	
0	0	Normal
0	1	Sleep
1	0	Stop

■ Oscillation Stabilization Wait Time

Because the oscillator stops in stop mode, an oscillation stabilization wait time is required when an oscillator starts operation.

One of three types of main clock oscillation stabilization wait times generated by the timebase timer can be selected as the oscillation stabilization wait time,.

In main clock mode, if the selected timebase timer interval time is less than the oscillation stabilization wait time, an interval timer interrupt request is issued during the oscillation stabilization wait time. Before entering stop mode in main clock mode, disable interrupt request output of the timebase timer (TBTC: TBIE = 0) if this is required.

3.8 Memory Access Mode

The only operating mode for MB89580B/BW memory access is single-chip mode.

■ Single-chip Mode

Single-chip mode uses only internal RAM and ROM. The CPU can access only the internal I/O, RAM, and ROM areas (internal access).

■ Mode Pins (MOD0, MOD1)

Be sure to set the mode pins (MOD0, MOD1) to V_{SS}.

For a reset, mode data and the reset vector are read from internal ROM.

Do not change the setting of the mode pins after the reset operation has terminated (during reset operation).

Table 3.8-1 "Mode Pin Settings" lists the settings for the mode pins.

Table 3.8-1 Mode Pin Settings

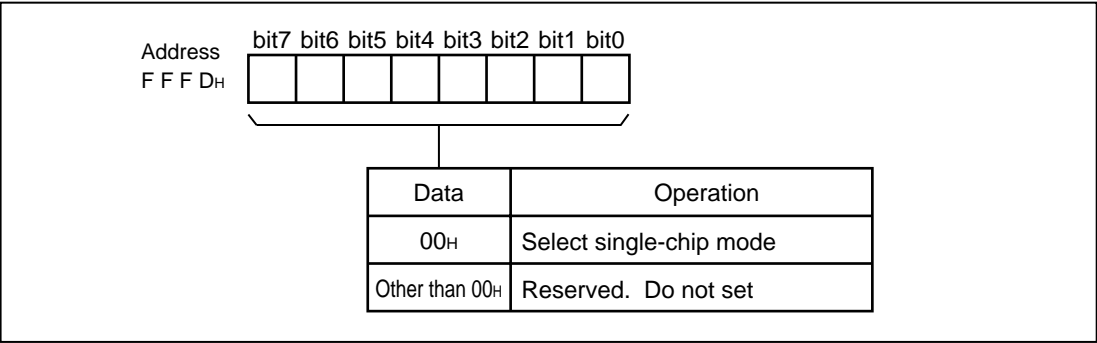
Pin state		Description
MOD0	MOD1	
V _{SS}	V _{SS}	Mode data and the reset vector are read from internal ROM.
V _{SS}	V _{CC}	Setting not allowed
V _{CC}	V _{SS}	
V _{CC}	V _{CC}	

■ Mode Data

Be sure to set mode data in internal ROM to 00H and select single-chip mode.

Figure 3.8-1 "Mode Data Configuration" shows the mode data configuration.

Figure 3.8-1 Mode Data Configuration



■ Memory Access Mode Selection Operation

Only single-chip mode can be selected.

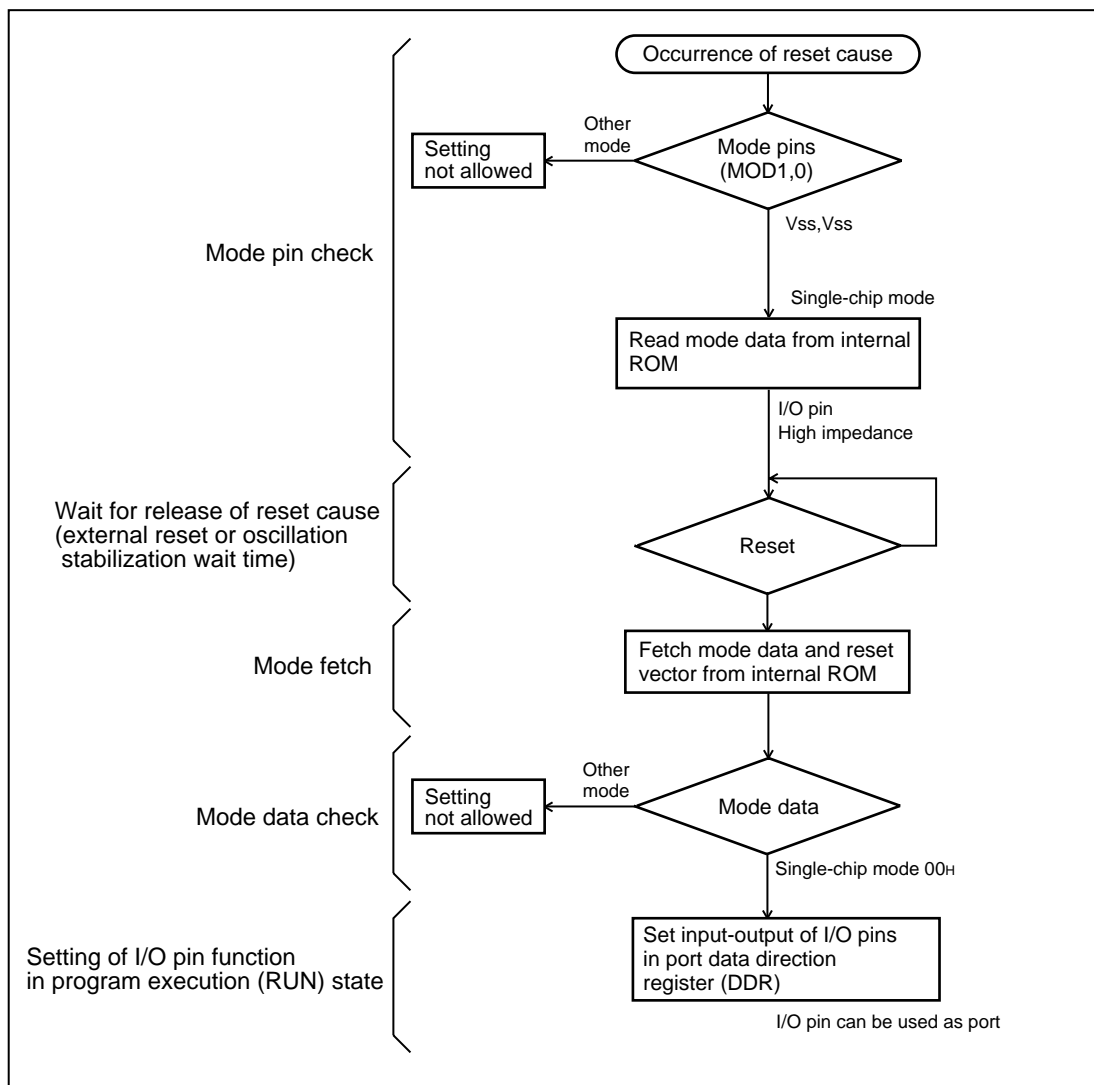
Table 3.8-2 "Mode Pins and Mode Data" lists the mode pins and mode data.

Table 3.8-2 Mode Pins and Mode Data

Memory access mode	Mode pins (MOD0, MOD1)	Mode data
Single-chip mode	V _{SS} , V _{SS}	00 _H
Other mode	Setting not allowed	Setting not allowed

Figure 3.8-2 "Memory Access Selection Operation" shows the memory access selection operation.

Figure 3.8-2 Memory Access Selection Operation



CHAPTER 4 I/O PORTS

This chapter explains the functions and operation of the I/O ports.

- 4.1 "Overview of the I/O Ports"
- 4.2 "Port 0"
- 4.3 "Port 1"
- 4.4 "Port 2"
- 4.5 "Port 3"
- 4.6 "Port 4"
- 4.7 "Port 5"
- 4.8 "Port 6"
- 4.9 "Program Example of the I/O Ports"

4.1 Overview of the I/O Ports

The I/O ports can be used as 6-port (53) general-purpose I/O ports.
Ports 3 to 6 can also be used for resources (I/O pins for peripheral functions).

■ I/O Port Functions

I/O ports have functions for outputting data from the CPU to the I/O pins and for fetching the signals input on the I/O pins to the CPU according to the setting of port data register (PDR). For some ports, the input-output direction of the I/O pin can be set at the bit level using the port data direction register (DDR).

The functions of the ports and the resources ports can be used for are listed below:

- Port 0: General-purpose I/O port
- Port 1: General-purpose I/O port
- Port 2: General-purpose output port
- Port 3: Both general-purpose I/O port and resources ports (external interrupt, clock output, and parallel port control pin)
- Port 4: Both general-purpose I/O port and resources ports (UART/SIO, PWM, parallel port data pin, and external FIFO data output)
- Port 5: Both general-purpose I/O port and resources ports (external FIFO control pin and parallel port control pin)
- Port 6: Both general-purpose input port and resources ports (external FIFO data input)

Table 4.1-1 "Functions of Ports" lists the functions of the ports. Table 4.1-2 "Port Registers" lists the port registers.

Table 4.1-1 Functions of Ports

Port name	Pin name	Input format	Output format	Function	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Port 0	P00 to P07	CMOS	CMOS push-pull	General-purpose I/O port	P07	P06	P05	P04	P03	P02	P01	P00
Port 1	P10 to P17			General-purpose I/O port	P17	P16	P15	P14	P13	P12	P11	P10
Port 2	P20 to P27			General-purpose output port	P27	P26	P25	P24	P23	P22	P21	P20
Port 3	P30/ $\overline{\text{INT0}}$ /CLK to P37/ $\overline{\text{INT7}}$	CMOS, resource interface with hysteresis		General-purpose I/O port	P37	P36	P35	P34	P33	P32	P32	P30
				External interrupt	$\overline{\text{INT7}}$	$\overline{\text{INT6}}$	$\overline{\text{INT5}}$	$\overline{\text{INT4}}$	$\overline{\text{INT3}}$	$\overline{\text{INT3}}$	$\overline{\text{INT2}}$	$\overline{\text{INT1}}$
				Resource 2	RDX	WEX	-	-	-	-	-	CLK
Port 4	P40/D0 to P47/D7/PWM	CMOS *1		General-purpose output port	P47	P46	P45	P44	P43	P42	P41	P40
				Parallel port output	D7	D6	D5	D4	D3	D2	D1	D0
				UART/SIO	-	UI	UO	UCK	-	-	-	-
				PWM	PWM2	PWM1	-	-	-	-	-	-
Port 5	P50/OBF/IBFX/W to P54/CEX	CMOS	CMOS *2 push-pull	General-purpose output port	-	-	-	P54	P53	P52	P51	P50
				Parallel port control	-	-	-	CEX	A0	-	-	OBF/IBFX
				External FIFO control	-	-	-	-	FFX	EFX	R	W
Port 6	P60/DIO to P67/DI7		-	General-purpose input port	P67	P66	P65	P64	P63	P62	P61	P60
				Parallel port input	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

*1: P44/UCK/D4 and P46/UI/D6 resources have an interface with hysteresis input.

*2: P52, P53, and P54 are set to N-ch open-drain output.

Table 4.1-2 Port Registers

Register name	Read, write	Address	Initial value
Port 0 data register (PDR0)	R/W	0000 _H	XXXXXXXX _B
Port 0 data direction register (DDR0) *	W	0001 _H	00000000 _B
Port 1 data register (PDR1)	R/W	0002 _H	XXXXXXXX _B
Port 1 data direction register (DDR1) *	W	0003 _H	00000000 _B
Port 2 data register (PDR2)	R/W	0004 _H	XXXXXXXX _B
Port 3 data register (PDR3)	R/W	000C _H	XXXXXXXX _B
Port 3 data direction register (DDR3)	R/W	000D _H	00000000 _B
Port 4 data register (PDR4)	R/W	0010 _H	XXXXXXXX _B
Port 4 data direction register (DDR4)	R/W	0011 _H	00000000 _B
Port 5 data register (PDR5)	R/W	0012 _H	XXX111XX _B
Port 5 data direction register (DDR5)	R/W	0013 _H	XXXXXX00 _B
Port 6 data register (PDR6)	R	0014 _H	XXXXXXXX _B

R/W: Readable and writable

R: Read only

W: Write only

X: Undefined

*: DDR0 and DDR1 cannot use bit manipulation commands.

4.2 Port 0

Port 0 is the general-purpose I/O port.

This section includes the port 0 configuration, pins, pin block diagram, and related registers.

■ Port 0 Configuration

Port 0 consists of the following three elements:

○ Port 0

- I/O pins (P00 to P07)
- Port 0 data register (PDR0)
- Port 0 data direction register (DDR0)

■ Port 0 Pins

Port 0 has eight CMOS I/O pins.

Table 4.2-1 "Port 0 Pins" lists the port 0 pins.

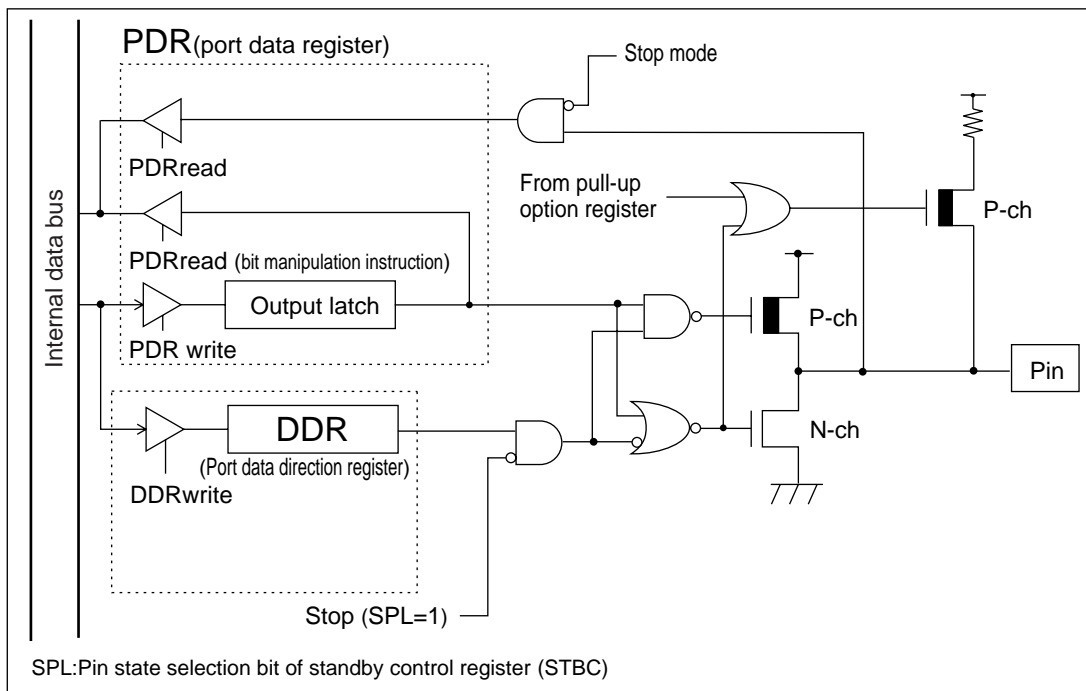
Table 4.2-1 Port 0 Pins

Port name	Pin name	Function	I/O		Circuit type
			Input	Output	
Port 0	P00	P00 general-purpose I/O	CMOS	CMOS	B
	P01	P01 general-purpose I/O			
	P02	P02 general-purpose I/O			
	P03	P03 general-purpose I/O			
	P04	P04 general-purpose I/O			
	P05	P05 general-purpose I/O			
	P06	P06 general-purpose I/O			
	P07	P07 general-purpose I/O			

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 0

Figure 4.2-1 Block Diagram of Port 0 Pins



■ Port 0 Registers (PDR0 and DDR0)

The registers related to port 0 are PDR0 and DDR0.

The bits in the registers correspond to the port 0 pins on a one-to-one basis.

Table 4.2-2 "Correspondence Between Port 0 Registers and Pins" shows the correspondence between port 0 registers and pins.

Table 4.2-2 Correspondence Between Port 0 Registers and Pins

Port name	Related register bit and corresponding pin								
Port 0	PDR0,DDR0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	P07	P06	P05	P04	P03	P02	P01	P00

4.2.1 Port 0 Registers (PDR0 and DDR0)

This section explains the registers related to port 0.

■ Functions of the Port 0 Registers

○ Port 0 data register (PDR0)

The PDR0 register indicates pin state. If a pin is set as an output port, the same value (0 or 1) as the output latch can be read, but if it is set as an input port, the output latch value cannot be read.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

○ Port 0 data direction register (DDR0)

The DDR0 register sets the input-output direction of the pin for the corresponding bit.

When the bit corresponding to the port is set to 1, the port is set as an output port. When the bit is set to 0, the port is set as an input port.

Table 4.2-3 Functions of the Port 0 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 0 data register (PDR0)	0	Pin state is L level.	Set the output latch to 0 and output the L level to the pin set as the output port.	R/W	0000 _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to 1 and output the H level to the pin set as the output port.			
Port 0 data direction register (DDR0)	0	Input port state	Disable output transistor operation and set input pin.	W	0001 _H	00000000 _B
	1	Output port state	Enable output transistor operation and set output pin.			

R/W: Readable and writable

W: Write only

X: Undefined

4.2.2 Operation of Port 0

This section explains the operation of port 0.

■ Operation of Port 0

○ Operation for output port

- When the corresponding DDR0 register bits are set to 1, port 0 becomes an output port.
- When port 0 is an output port, output transistor operation is enabled and output latch data is output to the pins.
- When data is written to the PDR0 register, data is retained by the output latch and output to the pins unchanged.
- When the PDR0 register is read, the pin values can be read.

○ Operation for input port

- When the corresponding DDR0 register bits are set to 0, port 0 becomes an input port.
- When port 0 is an input port, the output transistor is set to off and the pins enter the high-impedance state.
- When data is written to the PDR0 register, data is retained by the output latch, but it is not output to the pins.
- When the PDR0 register is read, the pin values can be read.

○ Operation for a reset

- When the CPU is reset, the DDR0 register values are initialized to 0. The output transistor is set to off (input port) and the pins enter the high-impedance state.
- The PDR0 register is not initialized by a reset. If port 0 is used as an output port, output data must be set in the PDR0 register before the corresponding DDR0 register is set to output.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, the pins enter the high-impedance state. This is because the output transistor is forcibly set to off regardless of the DDR0 register values. Input pin status is kept in a fixed level to prevent leakage from the pin when it is left in a open state.

Table 4.2-4 "Port 0 Pin States" lists the states of port 0 pins.

Table 4.2-4 Port 0 Pin States

Pin name	Normal operation	Stop (SPL = 1)	Reset
	Sleep Stop (SPL = 0)		
P00 to P07	General-purpose I/O port	Hi-Z	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register

Hi-Z: High impedance

○ Pull-up

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

4.3 Port 1

Port 1 is the general-purpose I/O port that also serves as input pins. This section includes the port 1 configuration, pins, pin block diagram, and related registers.

■ Port 1 Configuration

Port 1 consists of the following three elements:

- General-purpose I/O pins (P10 to P17)
- Port 1 data register (PDR1)
- Port 1 data direction register (DDR1)

■ Port 1 Pins

Port 1 has eight CMOS I/O pins.
Table 4.3-1 "Port 1 Pins" lists the port 1 pins.

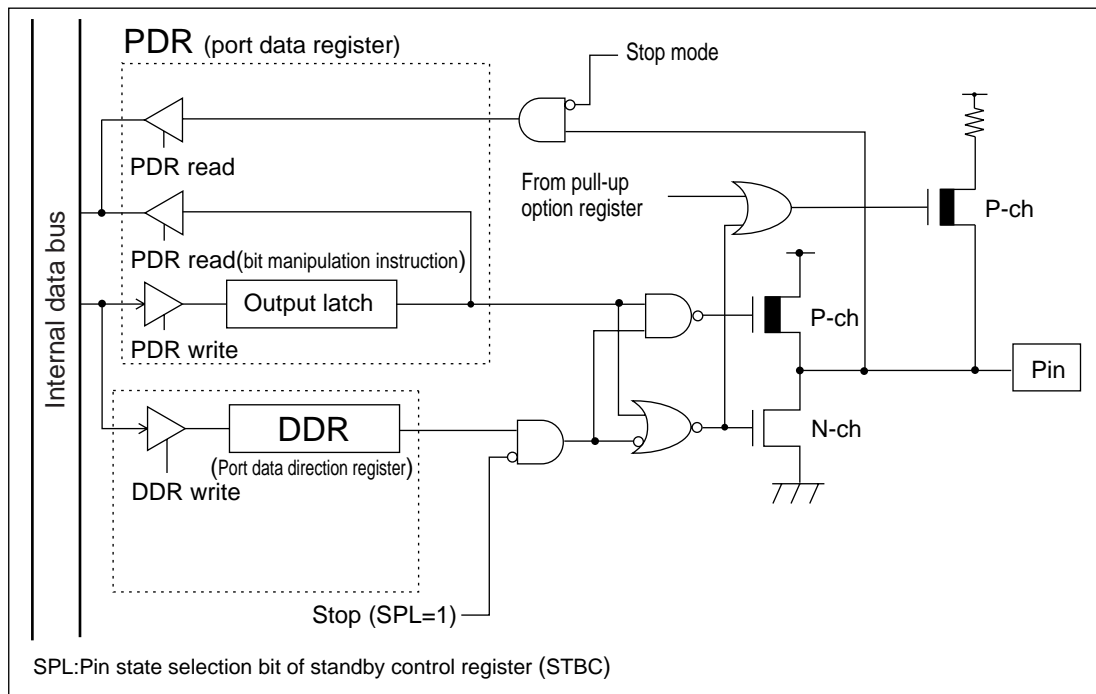
Table 4.3-1 Port 1 Pins

Port name	Pin name	Function	I/O		Circuit type
			Input	Output	
Port 1	P10	P10 general-purpose I/O	CMOS	CMOS	B
	P11	P11 general-purpose I/O			
	P12	P12 general-purpose I/O			
	P13	P13 general-purpose I/O			
	P14	P14 general-purpose I/O			
	P15	P15 general-purpose I/O			
	P16	P16 general-purpose I/O			
	P17	P17 general-purpose I/O			

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 1

Figure 4.3-1 Block Diagram of Port 1 Pin



■ Port 1 Registers (PDR1 and DDR1)

The registers related to port 1 are PDR1 and DDR1.

The bits in the registers correspond to the port 1 pins on a one-to-one basis.

Table 4.3-2 "Correspondence Between Port 1 Registers and Pins" shows the correspondence between port 1 registers and pins.

Table 4.3-2 Correspondence Between Port 1 Registers and Pins

Port name	Related register bit and corresponding pin								
Port 1	PDR1,DDR1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	P17	P16	P15	P14	P13	P12	P11	P10

4.3.1 Port 1 Registers (PDR1 and DDR1)

This section explains the registers related to port 1.

■ Functions of the Port 1 Registers

○ Port 1 data register (PDR1)

The PDR1 register indicates pin state. If a pin is set as an output port, the same value (0 or 1) as the output latch can be read, but if it is set as an input port, the output latch value cannot be read.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

○ Port 1 data direction register (DDR1)

The DDR1 register sets the input-output direction of the pin for the corresponding bit.

When the bits corresponding to the port are set to 1, the pins are set as an output port. When the bits are set to 0, the pins are set as an input port.

Table 4.3-3 "Functions of the Port 1 Registers" lists the functions of the port 1 registers.

Table 4.3-3 Functions of the Port 1 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 1 data register (PDR1)	0	Pin state is L level.	Set the output latch to 0 and output the L level to the pin set as the output port.	R/W	0002 _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to 1 and output the H level to the pin set as the output port.			
Port 1 data direction register (DDR1)	0	Input port state	Disable output transistor operation and set input pin.	W	0003 _H	00000000 _B
	1	Output port state	Enable output transistor operation and set output pin.			

R/W: Readable and writable

W: Write only

X: Undefined

4.3.2 Operation of Port 1

This section explains the operation of port 1.

■ Operation of Port 1

○ Operation for output port

- When the corresponding DDR1 register bits are set to 1, port 1 becomes an output port.
- When port 1 is an output port, output transistor operation is enabled and output latch data is output to the pins.
- When data is written to the PDR1 register, data is retained by the output latch and output to the pins unchanged.
- When the PDR1 register is read, the pin values can be read.

○ Operation for input port

- When the corresponding DDR1 register bits are set to 0, port 1 becomes an input port.
- When port 1 is an input port, the output transistor is set to off and the pins enter the high-impedance state.
- When data is written to the PDR1 register, data is retained by the output latch, but it is not output to the pins.
- When the PDR1 register is read, the pin values can be read.

○ Operation for a reset

- When the CPU is reset, the DDR1 register values are initialized to 0. The output transistor is set to off (input port) and the pins enter the high-impedance state.
- The PDR1 register is not initialized by a reset. If port 1 is used as an output port, output data must be set in the PDR1 register before the corresponding DDR1 register is set to output.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, pins enter the high-impedance state. This is because the output transistor is forcibly set to off regardless of the DDR1 register values. Input is fixed to prevent leak caused by opening input.

Table 4.3-4 "Port 1 Pin States" lists the states of the port 1 pins.

Table 4.3-4 Port 1 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P10 to P17	General-purpose I/O port	Hi-Z	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register
Hi-Z: High impedance

Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

4.4 Port 2

Port 2 is the general-purpose output port that also used for resource I/O. This section includes the port 2 configuration, pins, pin block diagram, and related registers.

■ Port 2 Configuration

Port 2 consists of the following two elements:

- General-purpose output pins (P20 to P27)
- Port 2 data register (PDR2)

■ Port 2 Pins

Port 2 has eight CMOS output pins.

Table 4.4-1 "Port 2 Pins" lists the port 2 pins.

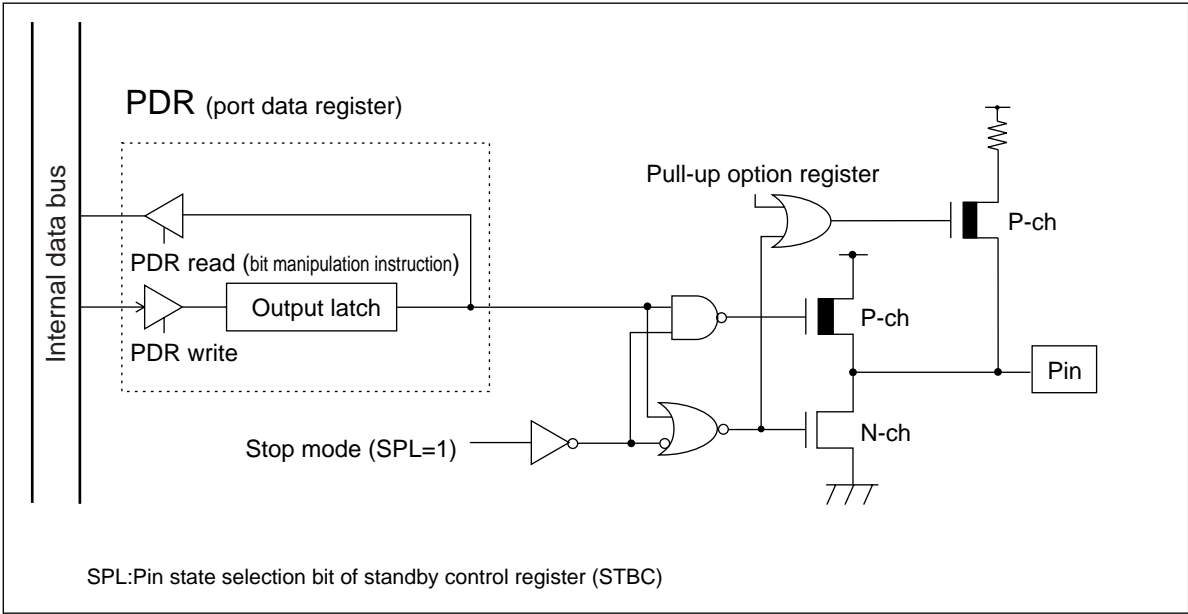
Table 4.4-1 Port 2 Pins

Port name	Pin name	Function	I/O		Circuit type
			Input	Output	
Port 2	P20	P20 general-purpose I/O	CMOS	CMOS	B
	P21	P21 general-purpose I/O			
	P22	P22 general-purpose I/O			
	P23	P23 general-purpose I/O			
	P24	P24 general-purpose I/O			
	P25	P25 general-purpose I/O			
	P26	P26 general-purpose I/O			
	P27	P27 general-purpose I/O			

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 2

Figure 4.4-1 Block Diagram of Port 2 Pins (P20 to P27)



■ Port 2 Register (PDR2)

The register related to port 2 is PDR2.

The bits in the register correspond to the port 2 pins on a one-to-one basis.

Table 4.4-2 "Correspondence Between Port 2 Registers and Pins" shows the correspondence between the port 2 register and pins.

Table 4.4-2 Correspondence Between Port 2 Registers and Pins

Port name	Related register bit and corresponding pin								
Port 2	PDR2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	P27	P26	P25	P24	P23	P22	P21	P20

4.4.1 Port 2 Register (PDR2)

This section explains the register related to port 2.

■ Function of Port 2 Register

○ Port 2 data register (PDR2)

The PDR2 register indicates the output latch state. The states of pins cannot be read.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

Table 4.4-3 "Functions of the Port 2 Registers" lists the function of port 2 register.

Table 4.4-3 Functions of the Port 2 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 2 data register (PDR2)	0	Pin state is L level.	Set the output latch to 0 and output the L level to the pin.	R/W	0004 _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to 1 and output the H level to the pin.			

R/W: Readable and writable

4.4.2 Operation of Port 2

This section explains the operation of port 2.

■ Operation of Port 2

○ Operation for output port

- When data is written to the PDR2 register, data is retained by the output latch and output to the pins via the output buffer.

○ Operation for a reset

- When the CPU is reset, the PDR2 register values are initialized to 0, and the pins enter the L level output state.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, the output transistor is forcibly set to off and the pins enter the high-impedance state. Input is fixed to prevent leak caused by opening input.

Table 4.4-4 "Port 2 Pin States" lists the states of the port 2 pins.

Table 4.4-4 Port 2 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P20 to P27	General-purpose output port	Hi-Z	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register

Hi-Z: High impedance

Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

4.5 Port 3

Port 3 is the general-purpose I/O port. Pins can be used by switching between resources and port for the corresponding bit.

This section explains the functions of the general-purpose I/O port.

This section includes the port 3 configuration, pins, pin block diagram, and related registers.

■ Port 3 Configuration

Port 3 consists of the following three elements:

- General-purpose I/O pins/resource I/O pins (P30/ $\overline{\text{INT0}}$ /CLK and P31/ $\overline{\text{INT1}}$ to P37/ $\overline{\text{INT7}}$)
- Port 3 data register (PDR3)
- Port 3 data direction register (DDR3)

■ Port 3 Pins

Port 3 has eight CMOS I/O pins.

Of these pins, a pin that is also used for resources cannot be used as a general-purpose I/O pin if resources are used.

Table 4.5-1 "Port 3 Pins" lists the port 3 pins.

Table 4.5-1 Port 3 Pins

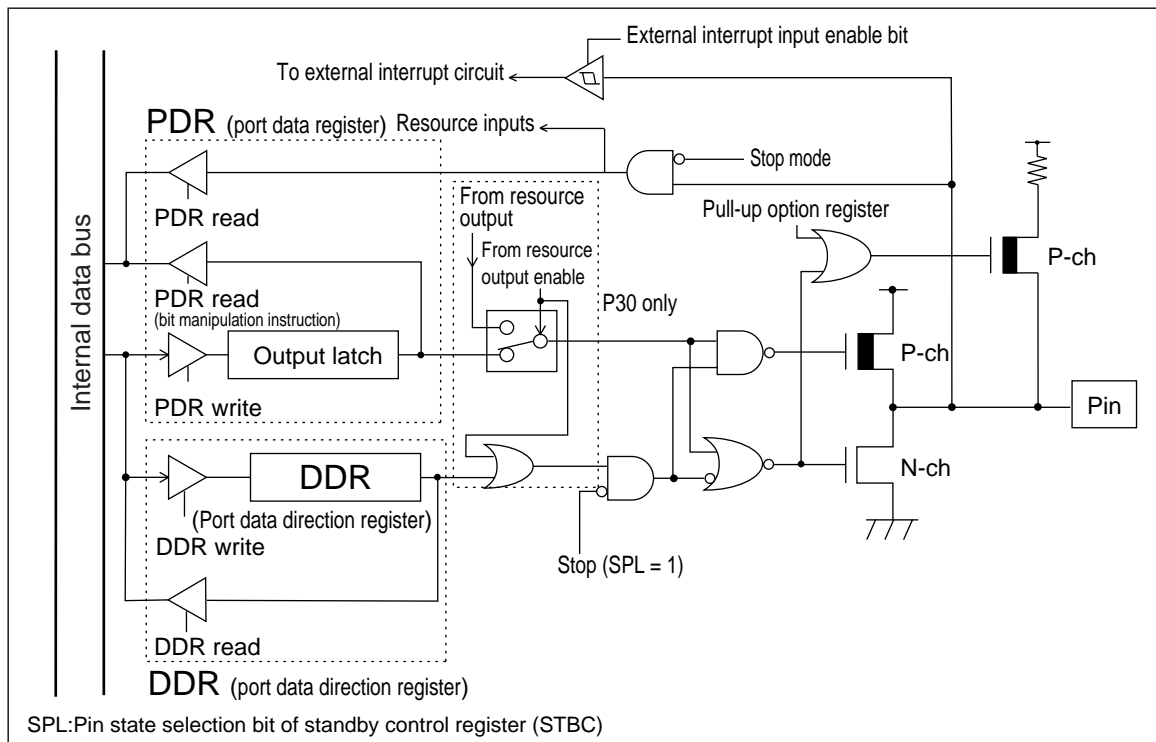
Port name	Pin name	Function	Resources	I/O		Circuit type
				Input	Output	
Port 3	P30/ $\overline{\text{INT0}}$ /CLK	P30 general-purpose I/O	External interrupt input ($\overline{\text{INT0}}$) CLK clock output	CMOS *	CMOS	E
	P31/ $\overline{\text{INT1}}$	P31 general-purpose I/O	External interrupt input ($\overline{\text{INT1}}$)			
	P32/ $\overline{\text{INT2}}$	P32 general-purpose I/O	External interrupt input ($\overline{\text{INT2}}$)			
	P33/ $\overline{\text{INT3}}$	P33 general-purpose I/O	External interrupt input ($\overline{\text{INT3}}$)			
	P34/ $\overline{\text{INT4}}$	P34 general-purpose I/O	External interrupt input ($\overline{\text{INT4}}$)			
	P35/ $\overline{\text{INT5}}$	P35 general-purpose I/O	External interrupt input ($\overline{\text{INT5}}$)			
	P36/ $\overline{\text{INT6}}$ /WEX	P36 general-purpose I/O	External interrupt input ($\overline{\text{INT6}}$) WEX parallel port control signal			
	P37/ $\overline{\text{INT7}}$ /RDX	P37 general-purpose I/O	External interrupt input ($\overline{\text{INT7}}$) RDX parallel port control signal			

*: Resource inputs are hysteresis.

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 3

Figure 4.5-1 Block Diagram of Port 3 Pin



■ Port 3 Registers (PDR3 and DDR3)

The registers related to port 3 are PDR3 and DDR3.

The bits in the registers correspond to the port 3 pins on a one-to-one basis.

Table 4.5-2 "Correspondence Between Port 3 Registers and Pins" shows the correspondence between port 3 registers and pins.

Table 4.5-2 Correspondence Between Port 3 Registers and Pins

Port name	Related register bit and corresponding pin								
	PDR3, DDR3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Port 3	Corresponding pin	P37	P36	P35	P34	P33	P32	P31	P30

4.5.1 Port 3 Registers (PDR3 and DDR3)

This section explains the registers related to port 3.

■ Functions of the Port 3 Registers

○ Port 3 data register (PDR3)

The PDR3 register indicates pin state. If a pin is set as an output port, the same value (0 or 1) as the output latch can be read, but if it is set as an input port, the output latch value cannot be read.

○ Port 3 data direction register (DDR3)

The DDR3 register sets the input-output direction of the pin for the corresponding bit.

When the bits corresponding to the port are set to 1, the pins are set as an output port. When the bits are set to 0, the pins are set as an input port.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

○ Setting for resource output

To use resources having output pins, set the resource output enable bit to enable.

Because resource output has priority, the setting values of the PDR3 and DDR3 registers relevant to the resource output pins are not effective to the resource output values or output enable state.

○ Setting for resource input

To use resources having input pins, set the pins corresponding to resource inputs as the input port. The corresponding output latch values are not effective.

Table 4.5-3 "Functions of the Port 3 Registers" lists the functions of the port 3 registers.

Table 4.5-3 Functions of the Port 3 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 3 data register (PDR3)	0	Pin state is L level.	Set the output latch to 0 and output the L level to the pin set as the output port.	R/W	000C _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to 1 and output the H level to the pin set as the output port.			
Port 3 data direction register (DDR3)	0	Input port state	Disable output transistor operation and set input pin.	R/W	000D _H	00000000 _H
	1	Output port state	Enable output transistor operation and set output pin.			

R/W: Readable and writable

4.5.2 Operation of Port 3

This section explains the operation of port 3.

■ Operation of Port 3

○ Operation for output port

- When the corresponding DDR3 register bits are set to 1, port 0 becomes an output port.
- When port 3 is an output port, output transistor operation is enabled and output latch data is output to the pins.
- When data is written to the PDR3 register, data is retained by the output latch and output to the pins unchanged.
- When the PDR3 register is read, the output latch values are always read.

○ Operation for input port

- When the corresponding DDR3 register bits are set to 0, port 3 becomes an input port.
- When port 3 is an input port, the output transistor is set to off and the pins enter the high-impedance state.
- When data is written to the PDR3 register, data is retained by the output latch, but it is not output to the pins.
- When the PDR3 register is read, the pin values can be read.

○ Operation for resource output

- When the resource output enable bit is set to enable, the corresponding pins are set to resource outputs.

○ Operation for resource input

- The DDR3 register bits corresponding to the resource input pins are set to 0 to set an input port.
- The pin values are always input as resource inputs.
- When the PDR3 register is read, the pin values can be read regardless of whether resources use input pins.

○ Operation for a reset

- When the CPU is reset, the DDR3 register values are initialized to 0. All output transistors are set to off (the port becomes an input port) and the pins enter the high-impedance state.
- The PDR3 register is not initialized by a reset. If port 3 is used as an output port, output data must be set in the PDR3 register before the corresponding DDR3 register is set to output.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, the output transistor is forcibly set to off and the pins enter the high-impedance state. Input is fixed to prevent leak caused by opening input.

Table 4.5-4 "Port 3 Pin States" lists the states of port 3 pins.

Table 4.5-4 Port 3 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P30 to P37	General-purpose I/O port/ resource I/O	Hi-Z (external interrupt input)	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register

Hi-Z: High impedance

Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

4.6 Port 4

Port 4 is the general-purpose I/O port that is also used for resource I/O. The pins can be used by switching between resources and port for each corresponding bit.

This section explains the functions of the general-purpose I/O port.

This section includes the port 0 configuration, pins, pin block diagram, and related registers.

■ Port 4 Configuration

Port 4 consists of the following three elements:

- General-purpose I/O pins/resource I/O pins (P40/D0 to P47/PWM2)
- Port 4 data register (PDR4)
- Port 4 data direction register (DDR4)

■ Port 4 Pins

Port 4 has eight CMOS I/O pins.

Of these pins, the pin that is also used for resources cannot be used as the general-purpose I/O port if resources are being used.

Table 4.6-1 "Port 4 Pins" lists the port 4 pins.

Table 4.6-1 Port 4 Pins

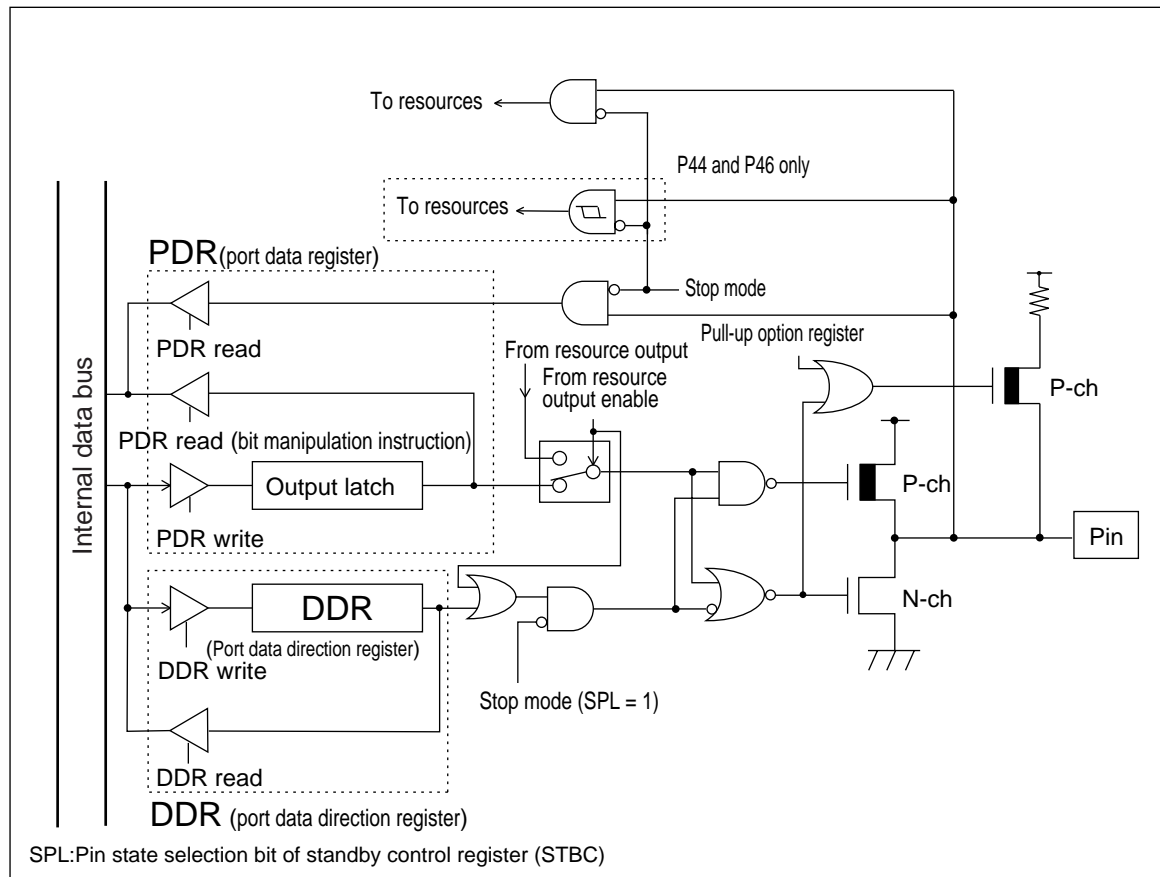
Port name	Pin name	Function	Resources	I/O		Circuit type
				Input	Output	
Port 4	P40/D0	P40 general-purpose I/O	D0 parallel port input-output/external FIFO data output	CMOS	CMOS	B
	P41/D1	P41 general-purpose I/O	D1 parallel port input-output/external FIFO data output			
	P42/D2	P42 general-purpose I/O	D2 parallel port input-output/external FIFO data output			
	P43/D3	P43 general-purpose I/O	D3 parallel port input-output/external FIFO data output			
	P44/D4/U _{CK}	P44 general-purpose I/O	D4 parallel port input-output/external FIFO data output U _{CK} UART/SIO	CMOS *		E
	P45/D5/U ₀	P45 general-purpose I/O	D5 parallel port input-output/external FIFO data output U ₀ UART/SIO	CMOS		B
	P46/D6/UI/PWM2	P46 general-purpose I/O	D6 parallel port input-output/external FIFO data output UIK UART/SIO PWM1 PWM output	CMOS *		E
	P47/D7/PWM2	P47 general-purpose I/O	D7 parallel port input-output/external FIFO data output PWM2 PWM output	CMOS		B

*: Resource interfaces are hysteresis inputs.

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 4

Figure 4.6-1 Block Diagram of Port 4 Pins (P40 to P47)



■ Port 4 Registers (PDR4 and DDR4)

The registers related to port 4 are PDR4 and DDR4.

The bits in the registers correspond to the port 4 pins on a one-to-one basis.

Table 4.6-2 "Correspondence Between Port 4 Registers and Pins" shows the correspondence between port 4 registers and pins.

Table 4.6-2 Correspondence Between Port 4 Registers and Pins

Port name	Related register bit and corresponding pin								
Port 4	PDR4, DDR4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	P47	P46	P45	P44	P43	P42	P41	P40

4.6.1 Port 4 Registers (PDR4 and DDR4)

This section explains the registers related to port 4.

■ Functions of the Port 4 Registers

○ Port 4 data register (PDR4)

The PDR4 register indicates pin state. If a pin is set as an output port, the same value (0 or 1) as the output latch can be read, but if it is set as an input port, the output latch value cannot be read.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

○ Port 4 data direction register (DDR4)

The DDR4 register sets the input-output direction of the pin for the corresponding bit.

When the bits corresponding to the port are set to 1, the pins are set as an output port. When the bits are set to 0, the pins are set as an input port.

○ Setting for resource output

To use resources having output pins, set the output enable bit of the resources to enable.

Because resource output has priority, the setting values of the PDR4 and DDR4 registers relevant to the resource inputs are not effective.

○ Setting for resource input

To use resources having input pins, set the pins corresponding to resource inputs as an input port. The corresponding latch values are not effective.

Table 4.6-3 "Functions of the Port 4 Registers" lists the functions of the port 4 registers.

Table 4.6-3 Functions of the Port 4 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 4 data register (PDR4)	0	Pin state is L level.	Set the output latch to 0 and output the L level to the pin set as the output port.	R/W	0010 _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to 1 and output the H level to the pin set as the output port.			
Port 4 data direction register (DDR4)	0	Input port state	Disable output transistor operation and set input pin.	R/W	0011 _H	00000000 _B
	1	Output port state	Enable output transistor operation and set output pin.			

R/W: Readable and writable

X: Undefined

4.6.2 Operation of Port 4

This section explains the operation of port 4.

■ Operation of Port 4

○ Operation for output port

- When the corresponding DDR4 register bits are set to 1, port 4 becomes an output port.
- When port 4 is an output port, output transistor operation is enabled and output latch data is output to the pins.
- When data is written to the PDR4 register, data is retained by the output latch and output to the pins unchanged.
- When the PDR4 register is read, the pin values can be read.

○ Operation for input port

- When the corresponding DDR4 register bits are set to 0, port 4 becomes an input port.
- When port 4 is an input port, the output transistor is set to off and the pins enter the high-impedance state.
- When data is written to the PDR4 register, data is retained by the output latch, but it is not output to the pins.
- When the PDR4 register is read, the pin values can be read.

○ Operation for resource output

- When the resource output enable bit is set to enable, the corresponding pins are set to resource output.
- When resource output is enabled, the pin values can be read by the PDR4 register and the resource output values can be read (except for parallel port output).

○ Operation for resource input

- The DDR4 register bits corresponding to the resource input pins are set to 0 to set the input port.
- The pin values are always input as resource inputs (except in stop mode).
- When the PDR4 register is read, the pin values can be read regardless of whether the resources use input pins.

○ Operation for a reset

- When the CPU is reset, the values of the DDR4 register bits are initialized to 0. The output transistor is set to off (input port) and the pins enter the high-impedance state.
- The PDR4 register is not initialized by a reset. If port 4 is used as an output port, output data must be set in the PDR4 register before the corresponding DDR4 register is set to output.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, pins enter the high-impedance state. This is because the output transistor is forcibly set to off regardless of the DDR4 register value. Input is fixed to prevent leak caused by opening input.

Table 4.6-4 "Port 4 Pin States" lists the states of port 4 pins.

Table 4.6-4 Port 4 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P40/D0 to P47/D7/ PWM2	General-purpose I/O port/ resource I/O	Hi-Z	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register

Hi-Z: High impedance

Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

Note:

The PDR4 and DDR4 registers in the general-purpose port also function as the PDCR parallel port data control register and the PSR parallel port status register. When the parallel port status register (PDIE bit of PSR) is used to switch the function from the general-purpose port, the registers hold the latest values.

4.7 Port 5

Port 5 is the general-purpose I/O port. The pins can be used by switching between resources and ports for each corresponding bit.

This section explains the functions of the general-purpose I/O port.

This section includes the port 0 configuration, pins, pin block diagram, and related registers.

■ Port 5 Configuration

Port 5 consists of the following three elements:

- General-purpose I/O pins/resource I/O pins (P50/OBF/IBFX/W to P54/CEX)
- Port 5 data register (PDR5)
- Port 5 data direction register (DDR5)

■ Port 5 Pins

Port 5 has five I/O pins for CMOS input and CMOS or N-ch open-drain output.

Of these pins, the pin that is also used for resources cannot be used as a general-purpose I/O port if resources are used.

Table 4.7-1 "Port 5 Pins" lists the port 5 pins.

Table 4.7-1 Port 5 Pins

Port name	Pin name	Function	Resources	I/O		Circuit type
				Input	Output	
Port 5	P50/OBF/IBFX/W	P50 general-purpose I/O	OBF/IBFX parallel port W external FIFO control signal	CMOS	CMOS	B
	P51/R	P51 general-purpose I/O	R external FIFO control signal			
	P52/EFX	P52 general-purpose I/O	EFX external FIFO control signal		N-ch open-drain	K
	P53/FFX/A0	P53 general-purpose I/O	FFX external FIFO control signal A0 parallel port			
	P54/CEX	P54 general-purpose I/O	CEX parallel port			

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 5

Figure 4.7-1 Block Diagram of Port 5 Pins P50 and P51

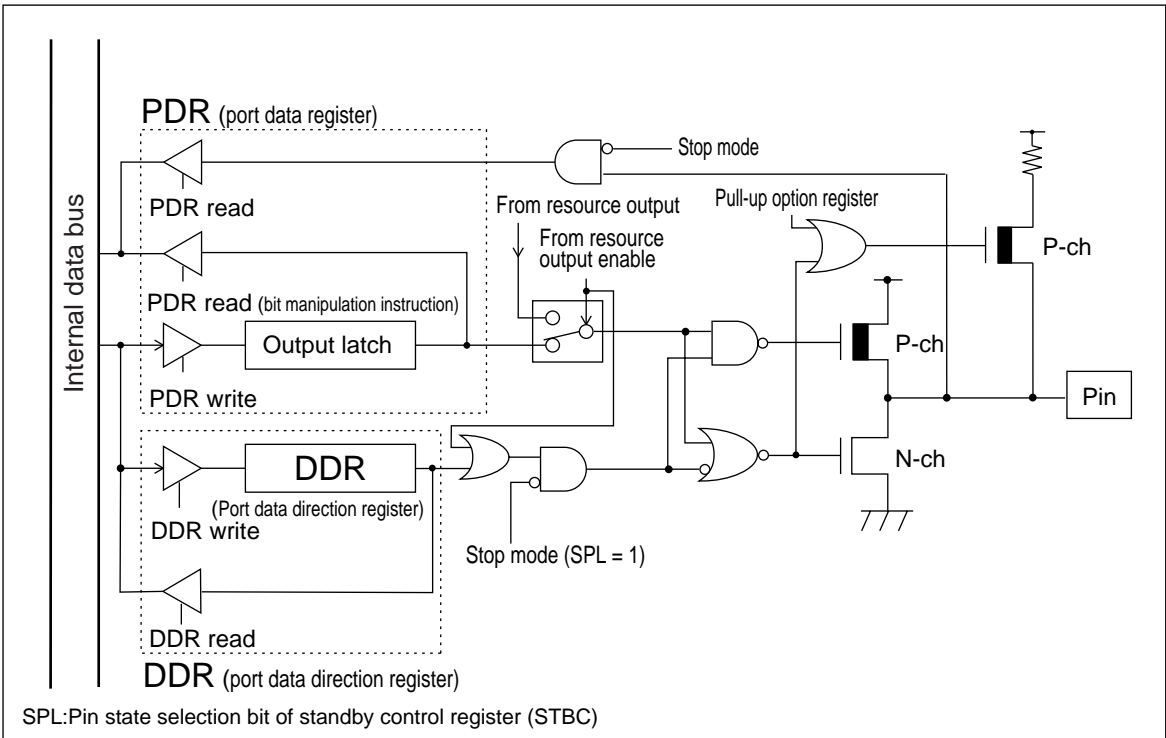
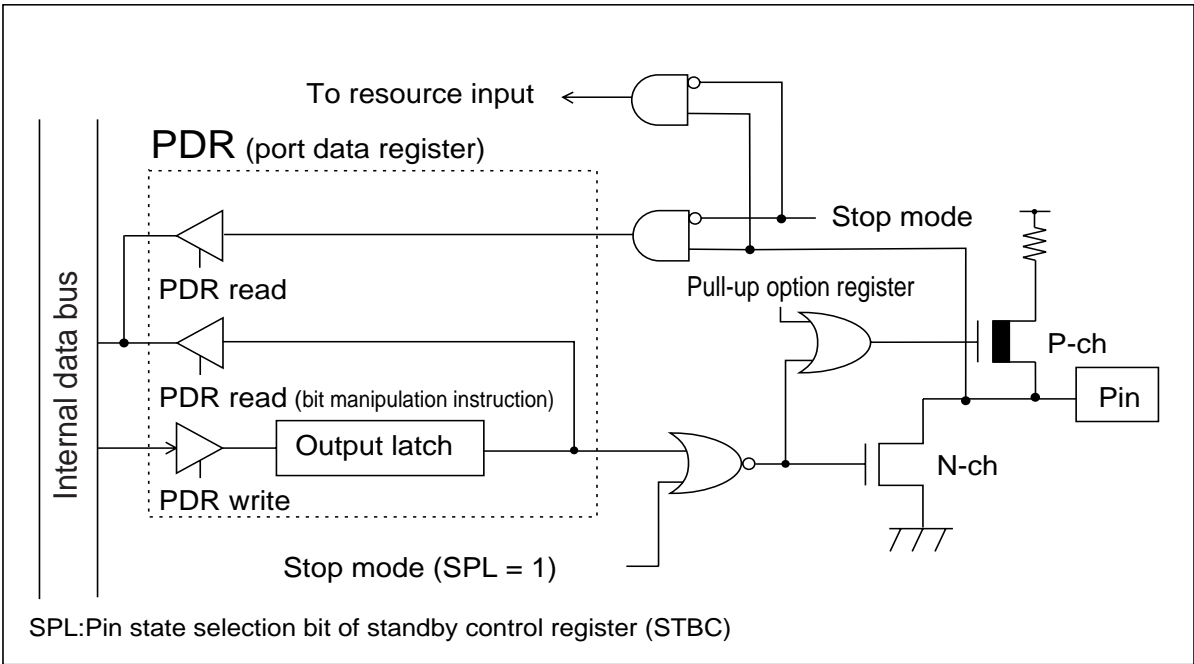


Figure 4.7-2 Block Diagram of Port 5 pins P52 to P54



■ Port 5 Registers (PDR5 and DDR5)

The registers related to port 5 are PDR5 and DDR5.

The bits in the registers correspond to the port 5 pins on a one-to-one basis.

Table 4.7-2 "Correspondence Between Port 5 Registers and Pins" shows the correspondence between port 5 registers and pins.

Table 4.7-2 Correspondence Between Port 5 Registers and Pins

Port name	Related register bit and corresponding pin								
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Port 5	PDR5								
	Corresponding pin	-	-	-	P54	P53	P52	P51	P50
	DDR5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	-	-	-	-	-	-	P51	P50

4.7.1 Port 5 Registers (PDR5 and DDR5)

This section explains the registers related to port 5.

■ Functions of the Port 5 Registers

○ Port 5 data register (PDR5)

The PDR5 register indicates pin state. If a pin is set as an output port, the same value (0 or 1) as the output latch can be read. However, the pin set for an input port does not enable reading of the output latch value.

Reference:

Because the bit manipulation instructions (SETB and CLRB) read the output latch values instead of the pin values, the output latch values of bits other than those that can be manipulated are not changed.

Note:

Set unused bits (bits 5 to 7) to 1.

○ Port 5 data direction register (DDR5)

The DDR5 register sets the input-output direction of the pin for the corresponding bit.

When the bits corresponding to the port are set to 1, the pins are set as an output port. When the bits are set to 0, the pins are set as an input port.

○ Setting for resource output

To use resources having output pins, set the resource output enable bit to enable.

Because resource output has priority, the setting values of the PDR5 and DDR5 registers relevant to the resource output pins are not effective to the resource output values and output enable state.

○ Setting for resource input

To use resources having input pins, set the pins corresponding to resource inputs to input port. The corresponding output latch values are not effective.

Table 4.7-3 "Functions of the Port 5 Registers" lists the functions of the port 5 registers.

Table 4.7-3 Functions of the Port 5 Registers

Register name	Data	Read	Write	Read, write	Address	Initial value
Port 5 data register (PDR5)	0	Pin state is L level.	Set the output latch to "0" and output the L level to the pin set as the output port.	R/W	0012 _H	XXXXXXXX _B
	1	Pin state is H level.	Set the output latch to "1" and output the H level to the pin set as the output port.			
Port 5 data direction register (DDR5)	0	Input port state	Disable output transistor operation and set input pin.	R/W	0013 _H	XXXXXX00 _B
	1	Output port state	Enable output transistor operation and set output pin.			

R/W: Readable and writable

X: Undefined

-: Unused

4.7.2 Operation of Port 5

This section explains the operation of port 5.

■ Operation of Port 5

○ Operation for output port

- When the corresponding DDR5 register bits, other than bits 2 to 4, are set to 1, port 5 becomes an output port.
- When port 5 is an output port, output transistor operation is enabled and output latch data is output to the pins.
- When data is written to the PDR5 register, data is retained by the output latch and output to the pins unchanged.
- When the PDR5 register is read, the pin values can be read.

○ Operation for input port

- When the corresponding DDR5 register bits, other than bits 2 to 4, are set to 0, port 5 becomes an input port.
- When port 5 is an input port, the output transistor is set to off and the pins enter the high-impedance state.
- When data is written to the PDR5 register, data is retained by the output latch, but it is not output to the pins.
- When the PDR5 register is read, the pin values can be read.

○ Operation for resource output

- When the resource output enable bit is set to enable, the corresponding pins are set to resource output.
- When resource output is enabled, the pin values can also be read by the PDR5 register and the resource output values can be read.

○ Operation for resource input

- The DDR5 register bits corresponding to resource input pins are set to 0 to set the input port.
- The pin values are always input as resource inputs (except in stop mode).
- When the PDR5 register is read, the pin values can be read regardless of whether the resources use input pins.

○ Operation for a reset

- When the CPU is reset, the values of the DDR5 register bits, other than bits 2 to 4, are initialized to 0. The output transistor is set to off (input port) and the pins enter the high-impedance state.
- The PDR5 register bits, other than bits 2 to 4, are not initialized by a reset. If port 5 is used as an output port, output data must be set in the PDR5 register before the corresponding DDR5 register is set to output.

○ Operation in stop mode

If the pin state selection bit (STBC: SPL) of the standby control register is set to 1 when stop mode is entered, pins enter the high-impedance state. This is because the output transistor is forcibly set to off regardless of the DDR5 register value. Input is fixed to prevent leak caused by opening input.

Table 4.7-4 "Port 5 Pin States" lists the states of the port 5 pins.

Table 4.7-4 Port 5 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P50/OBF/IBFX/W to P57/CEX2	General-purpose I/O port/ resource I/O	Hi-Z	Hi-Z

SPL: Pin state selection bit (STBC: SPL) of standby control register

Hi-Z: High impedance

Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

4.8 Port 6

Port 6 is the 3.3 V general-purpose input port. The pins can be used by switching between resources and port for each corresponding bit.

This section explains the functions of general-purpose input port.

This section includes the port 6 configuration, pins, pin block diagram, and related registers.

■ Port 6 Configuration

Port 6 consists of the following two elements:

- General-purpose input pins/resource input pins (P60/DI0 to P67/DI7)
- Port 6 data register (PDR6)

■ Port 6 Pins

Port 6 has eight CMOS input pins.

If resources are used, port 6 cannot be used as the general-purpose input port.

Table 4.8-1 "Port 6 Pins" lists the port 6 pins.

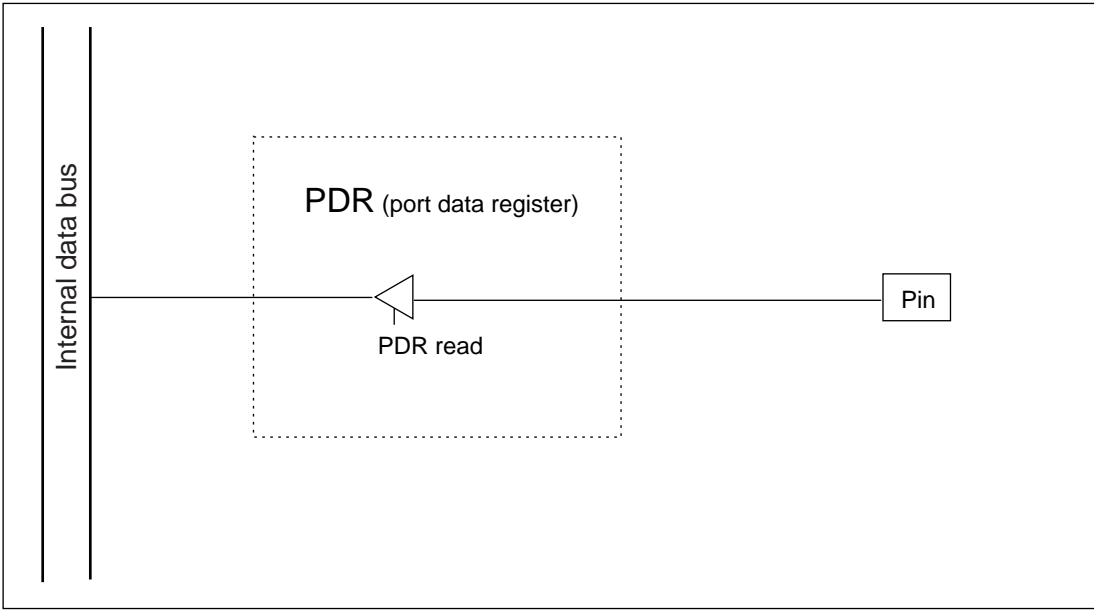
Table 4.8-1 Port 6 Pins

Port name	Pin name	Function	Resources	Input format	Circuit format
				Input	
Port 6	P60/DI0	P60 general-purpose input	External FIFO data input (DI0)	CMOS	F
	P61/DI1	P61 general-purpose input	External FIFO data input (DI1)		
	P62/DI2	P62 general-purpose input	External FIFO data input (DI2)		
	P63/DI3	P63 general-purpose input	External FIFO data input (DI3)		
	P64/DI4	P64 general-purpose input	External FIFO data input (DI4)		
	P65/DI5	P65 general-purpose input	External FIFO data input (DI5)		
	P66/DI6	P66 general-purpose input	External FIFO data input (DI6)		
	P67/DI7	P67 general-purpose input	External FIFO data input (DI7)		

See Section 1.7 "Pin Functions", for details of the circuit format.

■ Block Diagram of Port 6

Figure 4.8-1 Block Diagram of Port 6 Pin



■ Port 6 Register

The register related to port 6 is PDR6.

The bits in the register correspond to the port 6 pins on a one-to-one basis.

Table 4.8-2 "Correspondence Between Port 6 Registers and Pins" shows the correspondence between the port 6 register and pins.

Table 4.8-2 Correspondence Between Port 6 Registers and Pins

Port name	Related register bit and corresponding pin								
Port 6	PDR6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Corresponding pin	P67	P66	P65	P64	P63	P62	P61	P60

4.8.1 Port 6 Register (PDR6)

This section explains the register related to port 6.

■ Function of Port 6 Register

○ Port 6 data register (PDR6)

The PDR6 register indicates pin state. When the PDR6 register is read, the pin values can be read.

Table 4.8-3 "Functions of the Port 6 Registers" lists the functions of the port 6 register.

Table 4.8-3 Functions of the Port 6 Registers

Register name	Data	Read	Read, write	Address	Initial value
Port 6 data register (PDR6)	0	Pin state is "L" level.	R	0014 _H	XXXXXXXX _B
	1	Pin state is "H" level.			

R: Read only
X: Undefined

4.8.2 Operation of Port 6

This section explains the operation of port 6.

■ Operation of Port 6

○ Operation for input port

- When the PDR6 register is read, the pin values can be read.

○ Operation for resource input

- The pin values are always input as resource inputs.
- When the PDR6 register is read, the pin values can be read regardless of whether resources use input pins.

○ Operation for a reset

- The PDR6 register is not reset.

○ Operation in stop mode

Because the input being left open is liable to cause a leakage from the pin during stop mode, keep the pin in a fixed level at H or L.

Table 4.8-4 "Port 6 Pin States" lists the states of port 6 pins.

Table 4.8-4 Port 6 Pin States

Pin name	Normal operation Sleep Stop (SPL = 0)	Stop (SPL = 1)	Reset
P60/DI0 to P67/DI7	General-purpose input port/ resource input	Hi-Z	Hi-Z

Hi-Z: High impedance

4.9 Program Example of the I/O Ports

This section contains a program example that uses I/O ports.

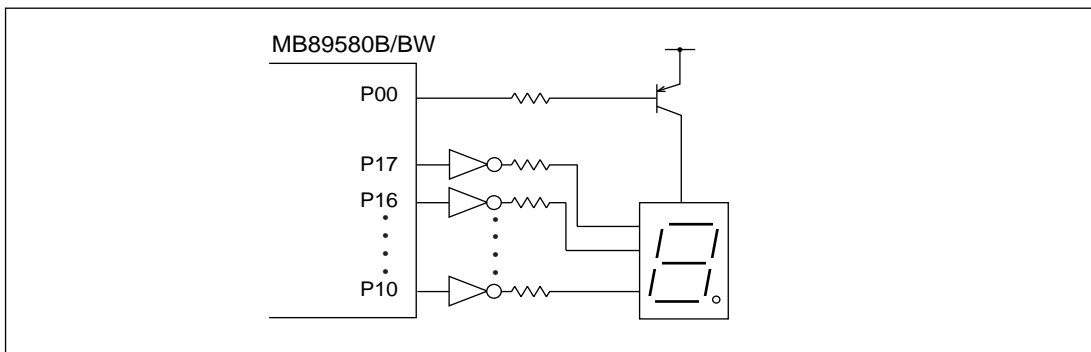
■ Program Example of the I/O Ports

○ Processing specification

- Ports 0 and 1 light all 7-segment (8-segment including Dp) LEDs.
- The P00 pin corresponds to the LED anode common pin, and the P10 to P17 pins correspond to segment pins.

Figure 4.9-1 "Example of Connecting an 8-Segment LED Connection" shows an example of connecting an 8-segment LED.

Figure 4.9-1 Example of Connecting an 8-Segment LED Connection



○ Coding example (comply with Softune V1)

```

PDR0 EQU 0000H ;Address port 0 data register
DDR0 EQU 0001H ;Address of port 0 data direction
               register
PDR1 EQU 0002H ;Address of port 1 data register
DDR1 EQU 0003H ;Address of port 1 data direction
               register

;---Main program-----
CSEG ; [CODE SEGMENT]
:
CLR   PDR0:0 ;Set P00 to L level
MOV   PDR1,#11111111B ;Set all bits for port 1 to H level
MOV   DDR0,#11111111B ;Set P00 to output. #xxxxxxx1B can
                     be set
MOV   DDR1,#11111111B ;Set all bits for port 1 to output
:
ENDS

;-----
END

```

CHAPTER 5 TIMEBASE TIMER

This chapter explains the functions and operation of the timebase timer.

- 5.1 "Overview of the Timebase Timer"
- 5.2 "Configuration of the Timebase Timer"
- 5.3 "Timebase Timer Control Register (TBTC)"
- 5.4 "Interrupts of the Timebase Timer"
- 5.5 "Operations of the Timebase Timer"
- 5.6 "Notes on Using the Timebase Timer"
- 5.7 "Program Example of the Timebase Timer"

5.1 Overview of the Timebase Timer

The timebase timer is a 21-bit free-run counter that is incremented by synchronization with an internal count clock (divide-by-two clock of the main clock oscillation). It has an interval timer function for choosing from among four time intervals. It also outputs an oscillation stabilization wait time and supplies operating clocks, such as a watchdog timer, to peripheral devices.

The timer is stops in any mode in which main clock oscillation stops.

■ Interval Timer Function

- This function generates repeated interrupts at specified intervals:
- The function generates an interrupt if an interval timer bit overflows in the timebase timer.
 - There are four interval timer bits (intervals) that can be selected.

Table 5.1-1 "Timebase Timer Intervals" lists these intervals.

Table 5.1-1 Timebase Timer Intervals

Internal count clock cycle	Interval
$2/F_{CH}(0.167\ \mu s)$	$2^{13}/F_{CH}$ (About 0.68 ms)
	$2^{15}/F_{CH}$ (About 2.73 ms)
	$2^{18}/F_{CH}$ (About 21.85 ms)
	$2^{22}/F_{CH}$ (About 349.53 ms)

F_{CH} : Main clock oscillation
Values in parentheses indicate main clock oscillation of 12 MHz.

■ Clock Supply Function

This function outputs an oscillation stabilization wait time (four types) and supplies operating clocks to some peripheral devices.

Table 5.1-2 "Clocks Supplied by Timebase Timer" lists the number of clock cycles supplied from the timebase timer.

Table 5.1-2 Clocks Supplied by Timebase Timer

Destination	Number of cycles	Remarks
Main clock oscillation stabilization wait time	$2^{14}/F_{CH}$ (About 1.37 ms)	Selected using the oscillation stabilization wait time selection bit (SYCC: WT1, WT0) in the system clock control register in the clock control section.
	$2^{17}/F_{CH}$ (About 10.92 ms)	
	$2^{18}/F_{CH}$ (About 21.85 ms)	
Watchdog timer	$2^{22}/F_{CH}$ (About 349.53 ms)	Count-up clock for watchdog timer

F_{CH} : Clock oscillation

Values in parentheses indicate main clock oscillation of 12 MHz.

Note:

The oscillation stabilization wait time is the standard because the oscillation cycle is unstable immediately after oscillation start.

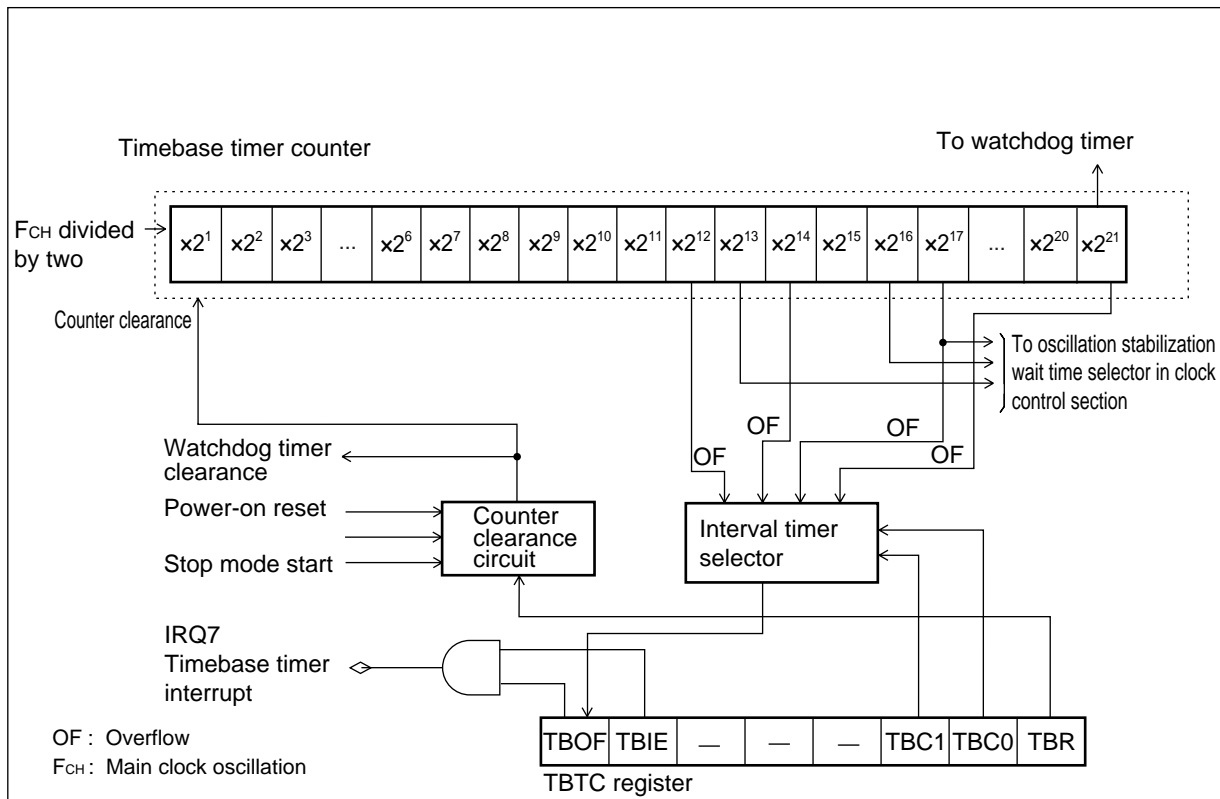
5.2 Configuration of the Timebase Timer

The timebase timer consists of the following four blocks:

- Timebase timer counter
- Counter clearance circuit
- Interval timer selector
- Timebase timer control register (TBTC)

■ Block Diagram of the Timebase Timer

Figure 5.2-1 Block Diagram of the Timebase Timer



○ Timer counter

This 21-bit counter is incremented using divide-by-two clocks of the main clock oscillation. The counter stops if the clock oscillation stops.

○ Counter clearance circuit

This circuit clears the counter if **TBR** is set to "0" in the **TBTC** register, the mode is switched to stop mode (**STP** = 1 in **STBC**), or a power-on reset occurs.

- **Interval timer selector**

This circuit selects one of the four interval timer bits in the timebase timer counter. Overflow of the selected bit causes an interrupt.

- **TBTC register**

This register is used to select an interval, control counter clearance and interrupts, and check conditions.

5.3 Timebase Timer Control Register (TBTC)

The timebase timer control register (TBTC) is used to select the interval time, clear the counter, control the interrupts, and check the status.

■ Timebase Timer Control Register (TBTC)

Figure 5.3-1 Timebase Timer Control Register (TBTC)

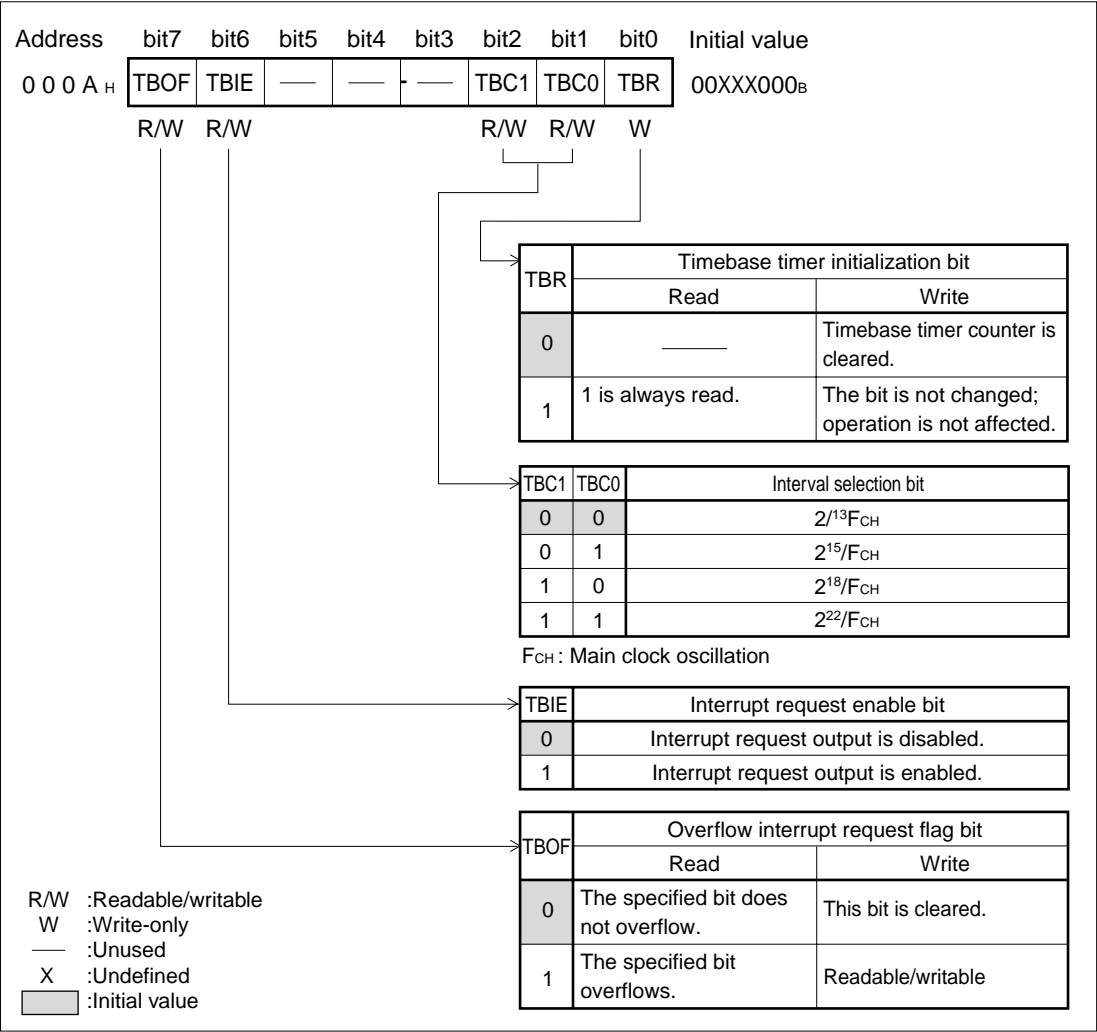


Table 5.3-1 Functions of TBTC Bits

Bit		Function
bit7	TBOF: Overflow interrupt request flag bit	<ul style="list-style-type: none"> This bit is set to "1" if the specified bit in the timebase timer counter overflows. An interrupt request is output when this bit and the interrupt request enable bit (TBIE) are set to "1". Writing "0" to this bit clears it. If "1" is written, the bit is not changed, and operation is not affected.
bit6	TBIE: Interrupt request enable bit	<ul style="list-style-type: none"> This bit disables and enables output of an interrupt request to the CPU. An interrupt request is output when this bit and TBOF are set to "1".
bit5 bit4 bit3	Unused	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit2 bit1	TBC1, TBC0: Interval selection bit	<ul style="list-style-type: none"> These bits select an interval timer cycle. An interval timer bit in the timebase timer counter is specified. Four intervals are selectable.
bit0	TBR: Timebase timer initialization bit	<ul style="list-style-type: none"> This bit clears the timebase timer counter. If "0" is written to this bit, the counter is cleared to "000000_H". If "1" is written, the bit is not changed, and operation is not affected. <p>Reference: "1" is always be read.</p>

5.4 Interrupts of the Timebase Timer

An interrupt is generated if the selected bit in the timebase timer counter overflows (interval timer function).

■ Interrupts Related to Operation of the Interval Timer Function

If the counter is incremented by an internal count clock and the specified interval timer bit overflows, the overflow interrupt request flag bit (TBOF in TBTC)) is set to "1". If the interrupt request enable bit (TBIE in TBTC) has also been set to "1", an interrupt request (IRQ7) is issued to the CPU. Clear the interrupt request with the interrupt handling routine by setting it to write "0" to the TBOF bit. The TBOF bit is set to "1" if the specified bit overflows regardless of the TBIE bit value.

Note:

- To enable an interrupt request output (TBIE = 1) after a reset is released, be sure to clear the TBOF bit (TBOF = 0).
- An interrupt request is issued immediately if interrupt request output is enabled (the TBIE bit changes from 0 to 1) while the TBOF bit is "1".
- The TBOF bit is not set to "1" if the counter is cleared (TBR = 0 in TBTC) and the selected bit overflows at the same time.

■ Oscillation Stabilization Wait Time and Timebase Timer Interrupt

If an interval shorter than the clock oscillation stabilization wait time is set, a timebase timer interval interrupt request (TBOF = 1 in TBTC) is issued when operation in clock mode starts. Disable timebase timer interrupts (TBIE = 0) when shifting to the mode in which clock oscillation stops (stop mode).

■ Register and Vector Table Related to Timebase Timer Interrupts

Table 5.4-1 Register and Vector Table Related to Timebase Timer Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bits to be set		High-order	Low-order
IRQ7	ILR2(007D _H)	L71(bit7)	L70(bit6)	FFEC _H	FFED _H

For the operation of interrupts, see Section 3.4.2 "Processing During Interrupt Operation".

5.5 Operations of the Timebase Timer

The timebase timer functions as interval timer and as a supplier of clocks to some peripheral devices.

■ Operation of the Interval Timer Function (Timebase Timer)

Figure 5.5-1 Setting for the Interval Timer Function



The timebase timer counter is incremented continuously by synchronization with the internal count clock (main clock oscillation) unless clock oscillation stops.

If the counter is cleared (TBR = 0), counting restarts at "0". When the selected interval timer bit overflows, the overflow interrupt request flag bit (TBOF) is set to "1". In other words, interrupt requests are issued at selected intervals by using the clearance time as the standard.

■ Operation of Clock Supply Function

The timebase timer is also used to set the clock oscillation stabilization wait time. The time until the bit for oscillation stabilization wait time overflows after the timebase timer counter is cleared is set as the oscillation stabilization wait time. An oscillation stabilization wait time can be selected from among three values by the using oscillation stabilization wait time selection bits in the system clock control register (WT1 and WT0 in SYCC).

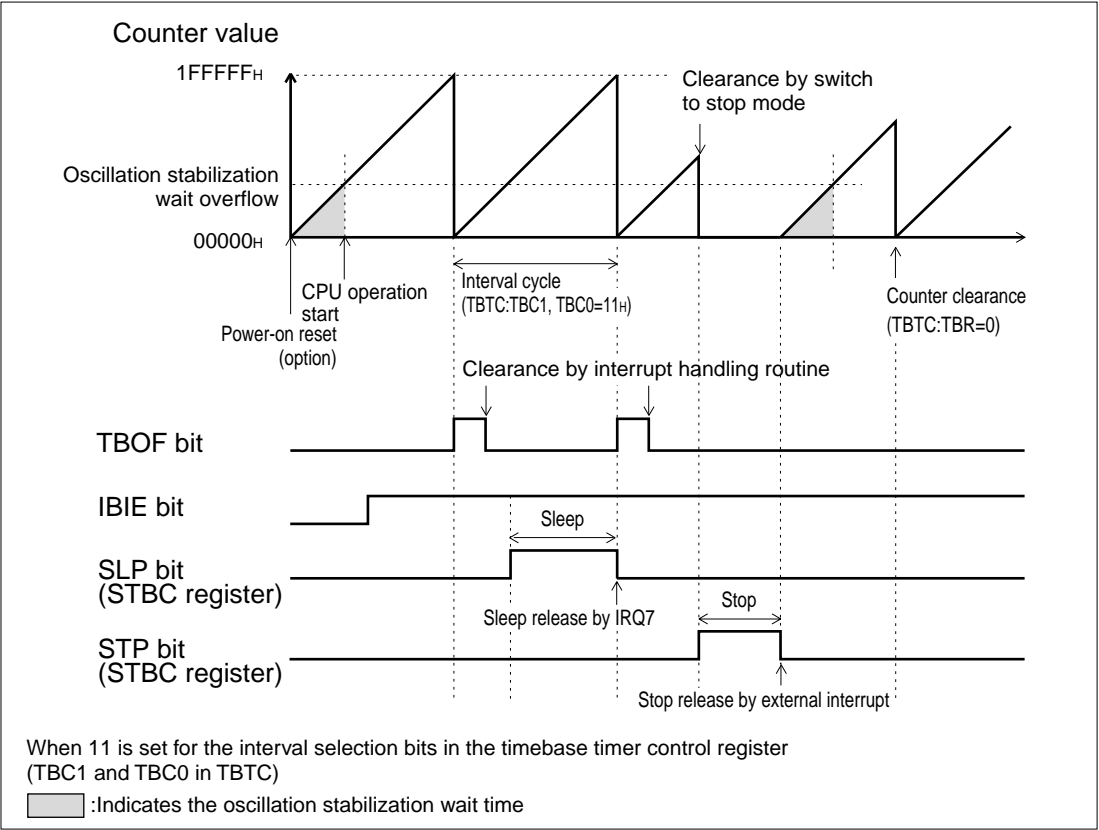
■ Operation of Timebase Timer

Figure 5.5-2 "Operation of Timebase Timer" shows the operations for the following events:

- A power-on reset occurs.
- The mode changes to sleep mode while the interval timer function is operating.
- The mode changes to stop mode.
- Clearing of the counter is requested.

When stop mode is set, the timebase timer is cleared and stopped. When processing is restored from stop mode, the timebase timer counts the oscillation stabilization wait time.

Figure 5.5-2 Operation of Timebase Timer



5.6 Notes on Using the Timebase Timer

This section provides notes on using the timebase timer.

■ Notes on Using the Timebase Timer

○ Notes on setting a program

Control cannot return from the interrupt handler if the interrupt request flag bit (TBOF in TBTC) is set to "1" and the interrupt request enable bit (TBIE in TBTC) is also set to "1". Be sure to clear the TBOF bit after setting the TBIE bit.

○ Timebase timer clearance

The timebase timer is also cleared when the clock oscillation stabilization wait time is required in addition to clearance by the timebase timer initialization bit (TBR = 0 in TBTC).

○ Use for oscillation stabilization wait time

The clock oscillation stabilization wait time is set using the timebase timer after the oscillator starts if power is turned on in stop mode because clock oscillation has stopped.

A suitable oscillation stabilization wait time should be selected for the type of a resonator connected to the clock oscillator (clock generation section).

See Section 3.6.5 "Oscillation Stabilization Wait Time" for details.

○ Peripheral functions to which clocks are supplied from timebase timer

In a mode in which clock oscillation has stopped, the counter is cleared and the timebase timer stops. The H level of a clock from the timebase timer may become shorter because of output after initialization if the counter is cleared. The L level may also become longer by as much as half its original length. The clock output to the watchdog timer is also initialized, but the watchdog timer operates with a normal cycle because its counter is also cleared.

5.7 Program Example of the Timebase Timer

This section contains a timebase timer program example.

■ Program Example of the Timebase Timer

○ Processing specifications

Interval timer interrupts are generated repeatedly at intervals of $2^{18}/F_{ch}$ (F_{ch} : Main clock oscillation (12 MHz)). This interval is about 21.85 ms (for 12 MHz).

○ Coding example (comply with Softune V1)

```

TBTC    EQU    0000AH                ;TBTC register address
TBOF    EQU    TBTC:7                ;TBOF bit definition
ILR2    EQU    007DH                ;Interrupt level setting
                                           register address
INT_V    DSEG    ABS                ;[DATA SEGMENT]
           ORG    0FFECH
IRQ7     DW     WARI                Interrupt vector setting
INT_V    ENDS
;-----Main program-----
CSEG                                           ;[CODE SEGMENT]
                                           ;The stack pointer (SP) has
                                           been initialized.

:
CLRI                                           ;Interrupt is disabled.
MOV     ILR2,#01111111B                ;Interrupt level setting
                                           (level 1)
MOV     TBTC,#01000100B                ;Interrupt request flag
                                           clearance, interrupt request
                                           output enabled
                                           218/Fch selection, timebase
                                           timer clearance
SETI                                           ;Interrupt program
:

;-----
WARI     CLRB     TBOF                ;Interrupt request flag
                                           clearance

PUSHW    A
XCHW     A,T
PUSHW    A
:
User Processing
:
POPW     A
XCHW     A,T
POPW     A
RETI
ENDS
;-----
END

```

CHAPTER 6 WATCHDOG TIMER

This chapter explains the functions and operation of the watchdog timer.

- 6.1 "Overview of the Watchdog Timer"
- 6.2 "Configuration of the Watchdog Timer"
- 6.3 "Watchdog Control Register (WDTC)"
- 6.4 "Operations of the Watchdog Timer"
- 6.5 "Notes on Using the Watchdog Timer"
- 6.6 "Program Example of the Watchdog Timer"

6.1 Overview of the Watchdog Timer

The watchdog timer is a one-bit counter that operates using output from the timebase timer as a count clock. It resets the CPU if it is not cleared within the specified time after it is activated.

■ Watchdog Timer Function

The watchdog timer is a counter that prevents a program from becoming a runaway. It must be cleared within the specified time after it is activated. If it is not cleared within the specified time due to, for example, an endless program loop, it generates a watchdog reset signal in a four-instruction cycle to the CPU.

Table 6.1-1 "Watchdog Timer Interval" lists the watchdog timer intervals. If the watchdog timer is not cleared within the specified time, a watchdog reset signal is generated between the minimum time and the maximum time in the table. Clear the counter within the minimum time.

Table 6.1-1 Watchdog Timer Interval

Count clock	
Timebase timer output (for main clock oscillation of 12 MHz)	
Minimum time	approx. 349.5 ms *
Maximum time	approx. 699.1 ms

*: Main clock oscillation (F_{CH}) divided by $2 \times$ count of the timebase timer (2^{21})

See Section 6.4 "Operations of the Watchdog Timer" for the minimum and maximum times.

Note:

- The watchdog timer counter is also cleared when the timebase timer is cleared ($TBR = 0$ in $TBTC$) because output from the timebase timer is selected as the count clock. The watchdog timer function is not valid if the timebase timer used as its count clock timer is cleared repeatedly within the interval specified for the watchdog timer.
- The watchdog timer counter is cleared when the mode changes to sleep or stop mode. It does not operate until ordinary operation (running status) is restored.
- The counter of the watchdog timer is cleared during USB data transfer and the timer resumes operation after the transfer is completed.

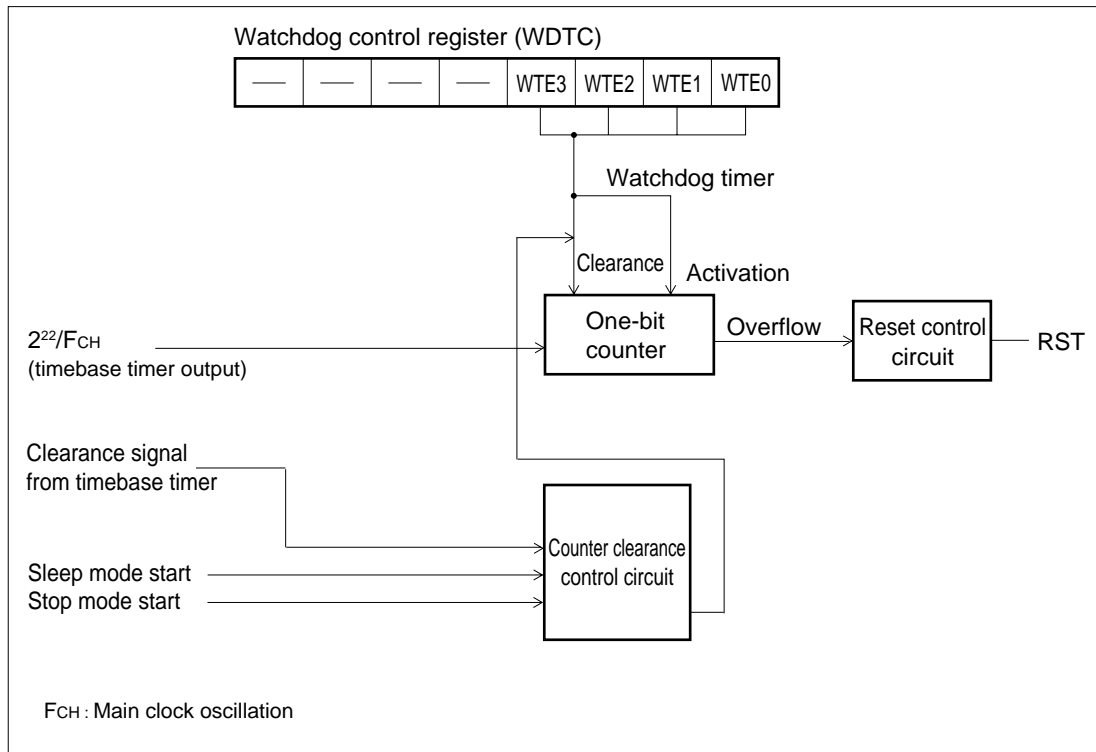
6.2 Configuration of the Watchdog Timer

The watchdog timer consists of the following four blocks:

- Watchdog timer counter
- Reset control circuit
- Counter clearance control circuit
- Watchdog control register (WDTC)

■ Block Diagram of the Watchdog Timer

Figure 6.2-1 Block Diagram of the Watchdog Timer



○ Watchdog timer counter (one-bit counter)

This one-bit counter operates using output from the timebase timer or watch prescaler as its count clock.

○ Reset control circuit

This circuit generates a reset signal to the CPU if the watchdog timer counter overflows.

○ Counter clearance control circuit

This circuit controls clearing of the watchdog timer counter and stopping of its operation.

○ Watchdog control register (WDTC)

This register selects a count clock, activates the watchdog timer counter, and clears it. Since the register can only be written to (write-only), bit manipulation instructions cannot be used on it.

6.3 Watchdog Control Register (WDTC)

The watchdog control register (WDTC) is a register to activate/clear the watchdog timer.

■ Watchdog Control Register (WDTC)

Figure 6.3-1 Watchdog Control Register (WDTC)

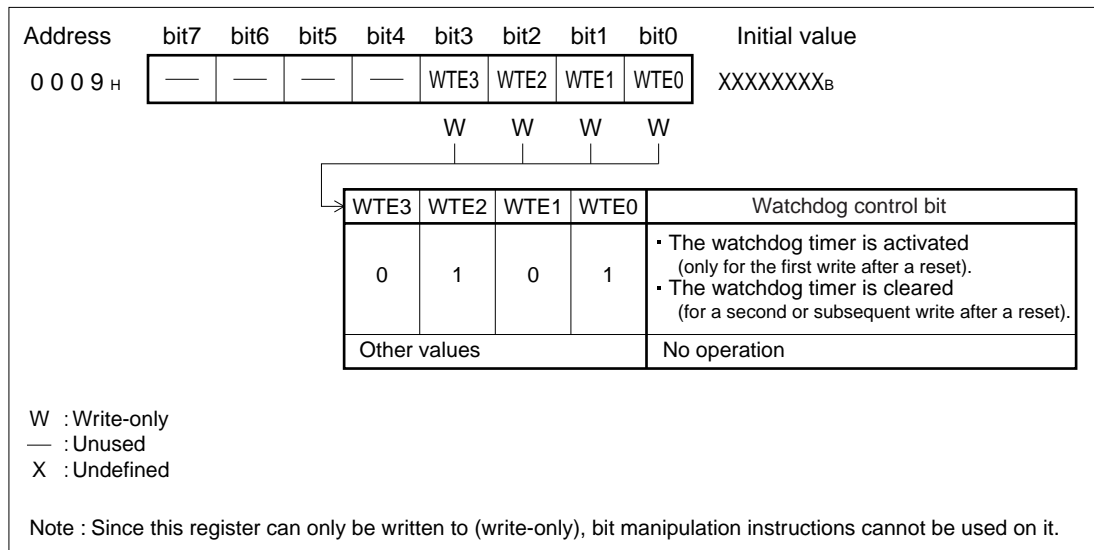


Table 6.3-1 Functions of WDTC Bits

Bit		Function
bit7	Unused	Always write 0.
bit6 bit5 bit4	Unused	<ul style="list-style-type: none"> Read values are undefined. Writing has no effect on operation.
bit3 bit2 bit1 bit0	WTE3, WTE2, WTE1, WTE0: Watchdog control bits	<ul style="list-style-type: none"> The watchdog timer is activated (the first write after a reset) or cleared (the second or subsequent write after a reset) when 0101_B is written. Writing a value other than 0101_B has no effect on operation. <p>Note: 1111_B is always read. Bit manipulation instructions cannot be used.</p>

6.4 Operations of the Watchdog Timer

The watchdog timer generates a watchdog reset signal if the watchdog timer counter overflows.

■ Operations of the Watchdog Timer

○ Activation of the watchdog timer

- The watchdog timer is activated when 0101_B is written to the watchdog control bits in the watchdog control register (WTE3 to 0 in WDTC) in the first write after a reset.
- The watchdog timer, once it is activated, can be stopped only by a reset.

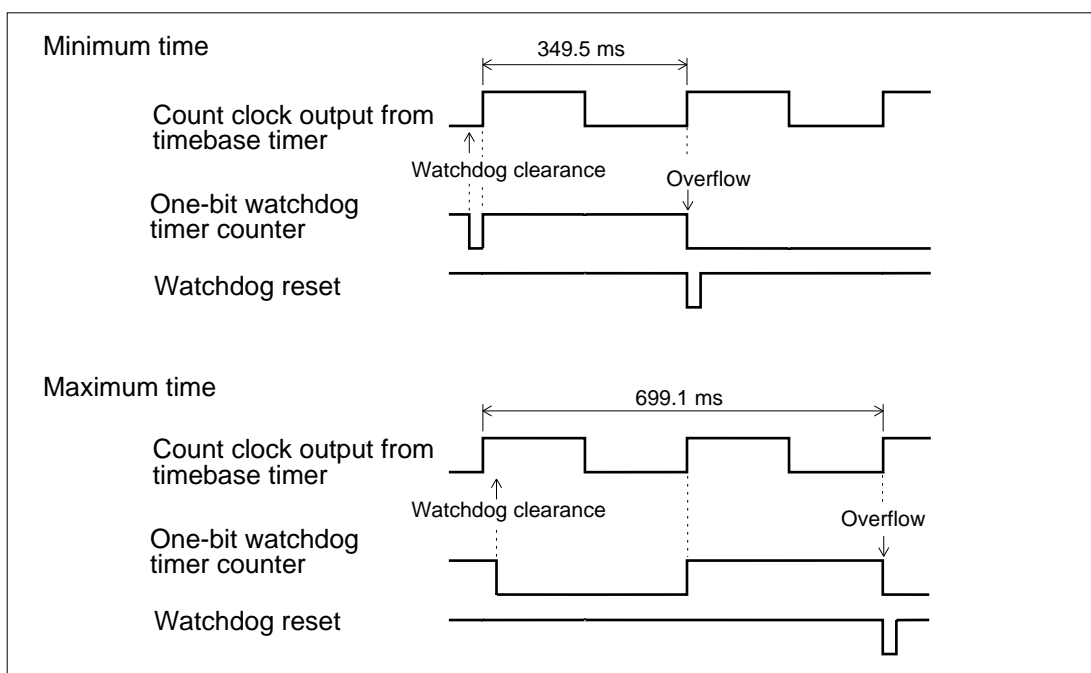
○ Clearance of the watchdog timer

- The watchdog timer counter is cleared when 0101_B is written to the watchdog control bits in the watchdog control register (WTE3 to 0 in WDTC) in the second or subsequent write after a reset.
- If the counter is not cleared within the specified time, it overflows and an internal reset signal in a four-instruction cycle is generated.

○ Watchdog timer interval

The interval changes according to watchdog timer clearance timing. Figure 6.4-1 "Watchdog Timer Clearance and Interval" shows the relationship between the clearance timing and interval with output from the timebase timer used as the count clock (for Main clock oscillation of 12MHz).

Figure 6.4-1 Watchdog Timer Clearance and Interval



6.5 Notes on Using the Watchdog Timer

This section provides notes on using the watchdog timer.

■ Notes on Using the Watchdog Timer

○ Stopping of the watchdog timer

The watchdog timer, once it is activated, cannot be stopped until a reset occurs.

○ Clearance of the watchdog timer

- If a counter (timebase timer) used to generate watchdog timer count clocks is cleared, the watchdog timer counter is also cleared.
- The watchdog timer counter is cleared if the mode is changed to sleep or stop mode.

○ Program creation

If a program is created so that the watchdog timer is cleared repeatedly in the main loop, the main loop processing time, including interrupt handling, must be within the minimum interval.

6.6 Program Example of the Watchdog Timer

This section shows a program example.

■ Program Example of the Watchdog Timer

○ Processing specifications

- The watchdog timer is activated immediately after the program starts.
- The watchdog timer is cleared in each loop of the main program.
- The cycle of the main loop, including interrupt handling, must be within the minimum interval (about 349.5 ms for 12 MHz).

○ Coding example (comply with Softune V1)

```

WDTC      EQU    00009H          ;Address of WDTC register
WDT_CLR   EQU    00000101B

VECT       DSEG  ABS             ;[DATA SEGMENT]
          ORG    0FFFEH
RST_V     DW     PROG           ;Reset vector setting
VECT       ENDS

;-----Main program-----
          CSEG                  ;[CODE SEGMENT]
PROG       ;Initialization routine for a reset
          MOVW   SP,#0280H       ;Setting of stack pointer initial
                                   (value for interrupt handling)
          :
          Peripheral function (interrupt) initialization
          :
INIT       MOV    WDTC,#WDT_CLR  ;Watchdog timer activation
          :
MAIN       MOV    WDTC,#WDT_CLR  ;Watchdog timer clearance
          :
          User processing (Interrupt handling may be activated
          durinthis time.)
          :
          JMP     MAIN           ;Must be set in a loop with a cycle
                                   shorter than the minimum interval.
          ENDS

;-----
          END

```

CHAPTER 7 2-CHANNEL 8-BIT PWM TIMER

This chapter explains the functions and operation of the 2-channel 8-bit PWM timer.

- 7.1 "Overview of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)"
- 7.2 "Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)"
- 7.3 "Configuration of the 2-Channel 8-Bit PWM Timer"
- 7.4 "Pins of the 2-Channel 8-Bit PWM Timer"
- 7.5 "Registers of the 2-Channel 8-Bit PWM Timer"
- 7.6 "Interrupts of the 2-Channel 8-Bit PWM Timer"
- 7.7 "Operations of the Interval Timer Function"
- 7.8 "Operations of the 8-Bit PWM Mode"
- 7.9 "Operations of the 7-Bit PWM Mode"
- 7.10 "Operations of the CH12PWM Mode"
- 7.11 "Prescaler Operation of the 2-Channel 8-Bit PWM Timer"
- 7.12 "Operation for Standby Mode and Intermediate Stop"
- 7.13 "Notes on Using the 2-Channel 8-Bit PWM Timer"
- 7.14 "Program Examples of the 2-Channel 8-Bit PWM Timer (Interval Function)"
- 7.15 "Program Examples of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)"

7.1 Overview of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)

The 2-channel 8-bit PWM timer consists of two 8-bit PWM timers (CH1, CH2) that are incremented by synchronization with internal count clocks that can be selected from among four types. Each of these 8-bit PWM timers can be selected as an interval timer that can output rectangular waves or a PWM timer with 8-bit or 7-bit resolution. The interval timer function can operate in 8-bit timer mode, where CH1 and CH2 are used independently, and in CK12 mode, where CH1 and CH2 are linked for use.

■ Interval Timer Function (Rectangular Output Function)

This function generates interrupts repeatedly at selected intervals.

Since the output level of a pin (PWM pin) can also be inverted each time an interrupt is generated, rectangular output with an optional frequency is possible.

○ 8-bit timer mode

The CH1 and CH2 timers can operate independently in this mode:

- Interval timer operation with a cycle between the count clock cycle and a cycle obtained by multiplying it by 2^8 is possible.
- A count clock can be selected from among four clocks.

Table 7.1-1 "Intervals and Rectangular Wave Output Ranges" lists intervals and rectangular output ranges.

Table 7.1-1 Intervals and Rectangular Wave Output Ranges

Count clock cycle		Interval	Rectangular wave output (Hz)
Internal count clock	$1t_{\text{inst}}$	$1t_{\text{inst}}$ to 2^8t_{inst}	$1/(2t_{\text{inst}})$ to $1/(2^9t_{\text{inst}})$
	$8t_{\text{inst}}$	2^3t_{inst} to $2^{11}t_{\text{inst}}$	$1/(2^4t_{\text{inst}})$ to $1/(2^{12}t_{\text{inst}})$
	$16t_{\text{inst}}$	2^4t_{inst} to $2^{12}t_{\text{inst}}$	$1/(2^5t_{\text{inst}})$ to $1/(2^{13}t_{\text{inst}})$
	$64t_{\text{inst}}$	2^6t_{inst} to $2^{14}t_{\text{inst}}$	$1/(2^7t_{\text{inst}})$ to $1/(2^{15}t_{\text{inst}})$

t_{inst} : Instruction cycle (depending on the clock mode)

○ Example of calculation of the interval and rectangular output frequency

The interval can be calculated as shown below when the count clock cycle is set to $1 t_{\text{inst}}$ and main clock oscillation (F_{CH}) has been set to 12 MHz and the PWM compare register (COMR) value has been set to DD_H (221). The frequency of a rectangular wave output from the PWM pin in continuous operation without this COMR value being changed can be calculated as shown below.

It is assumed in this example that the fastest clock is selected ($\text{CS1/CS0} = 11_\text{B}$, one instruction cycle = $4/F_{\text{CH}}$) as the clock mode ($\text{SCS} = 1$) in the system clock control register (SYCC).

7.1 Overview of the 2-Channel 8-Bit PWM Timer (Interval Timer Function)

$$\begin{aligned}\text{Interval} &= (1 \times 4/F_{CH}) \times (\text{COMR register value} + 1) \\ &= (4/12 \text{ MHz}) \times (221 + 1) \\ &= 74 \mu\text{s}\end{aligned}$$

$$\begin{aligned}\text{Output frequency} &= F_{CH}/(1 \times 8 \times (\text{COMR register value} + 1)) \\ &= 12 \text{ MHz} / (8 \times (221 + 1)) \\ &\text{nearby equal to } 6.76 \text{ kHz}\end{aligned}$$

○ CK12 mode

In this mode, the CH1 and CH2 timers are used in linkage. Rectangular output from the CH1 timer becomes the count clock for the CH2 timer in this mode:

- For the CH1 and CH2 timers, the interval timer can operate with a cycle between the count clock cycle and a cycle obtained by multiplying it by 2^8 .
- The CH1 timer is selected from among four count clock types.
- Rectangular output from the CH1 timer is used as the count clock for the CH2 timer.

Table 7.1-2 "Intervals and Rectangular Wave Output Ranges" lists intervals and rectangular output ranges.

Table 7.1-2 Intervals and Rectangular Wave Output Ranges

Count clock cycle			Interval	Rectangular wave output (Hz)
CH1	Internal count clock	$1t_{\text{inst}}$	$1t_{\text{inst}}$ to 2^8t_{inst}	$1/(2t_{\text{inst}})$ to $1/(2^9t_{\text{inst}})$
		$8t_{\text{inst}}$	2^3t_{inst} to $2^{11}t_{\text{inst}}$	$1/(2^4t_{\text{inst}})$ to $1/(2^{12}t_{\text{inst}})$
		$16t_{\text{inst}}$	2^4t_{inst} to $2^{12}t_{\text{inst}}$	$1/(2^5t_{\text{inst}})$ to $1/(2^{13}t_{\text{inst}})$
		$64t_{\text{inst}}$	2^6t_{inst} to $2^{14}t_{\text{inst}}$	$1/(2^7t_{\text{inst}})$ to $1/(2^{15}t_{\text{inst}})$
CH2	CH1 rectangular output	$2t_{\text{inst}}$ to $2^{15}t_{\text{inst}}$	$2t_{\text{inst}}$ to $2^{23}t_{\text{inst}}$	$1/(2^2t_{\text{inst}})$ to $1/(2^{24}t_{\text{inst}})$

t_{inst} : Instruction cycle (depending on the clock mode)

○ Example of calculation of the interval and rectangular output frequency

The interval can be calculated as shown below when the CH1 count clock cycle is set to 1 t_{inst} and main clock oscillation (F_{CH}) has been set to 12 MHz and DD_H (221) has been set in PWM compare registers 1 and 2 (COMR1, COMR2). The frequencies of rectangular waves output from the PWM pin in continuous operation without any changes to these COMR values can be calculated as shown below.

It is assumed in this example that the fastest clock is selected ($CS1/CS0 = 11_B$, one instruction cycle = $4/F_{CH}$) as clock mode ($SCS = 1$) in the system clock control register (SYCC).

$$\begin{aligned}
 \text{CH1 interval} &= (1 \times 4/F_{CH}) \times (\text{COMR1 register value} + 1) \\
 &= (4/12 \text{ MHz}) \times (221 + 1) \\
 &= 74 \mu\text{s} \\
 \text{CH1 output frequency} &= F_{CH}/(1 \times 8 \times (\text{COMR1 register value} + 1)) \\
 &= 12 \text{ MHz} / (8 \times (221 + 1)) \\
 &\text{ nearly equal to } 6.76 \text{ kHz} \\
 \text{CH2 interval} &= (1/\text{CH1 output frequency}) \times (\text{COMR2 register value} + 1) \\
 &= (1/6.76 \text{ kHz}) \times (221 + 1) \\
 &\text{ nearly equal to } 32.8 \text{ ms} \\
 \text{CH2 output frequency} &= \text{CH1 output frequency}/(\text{COMR2 register value} + 1) \\
 &= 6.76 \text{ kHz} / (221 + 1) \\
 &\text{ nearly equal to } 30.45 \text{ Hz}
 \end{aligned}$$

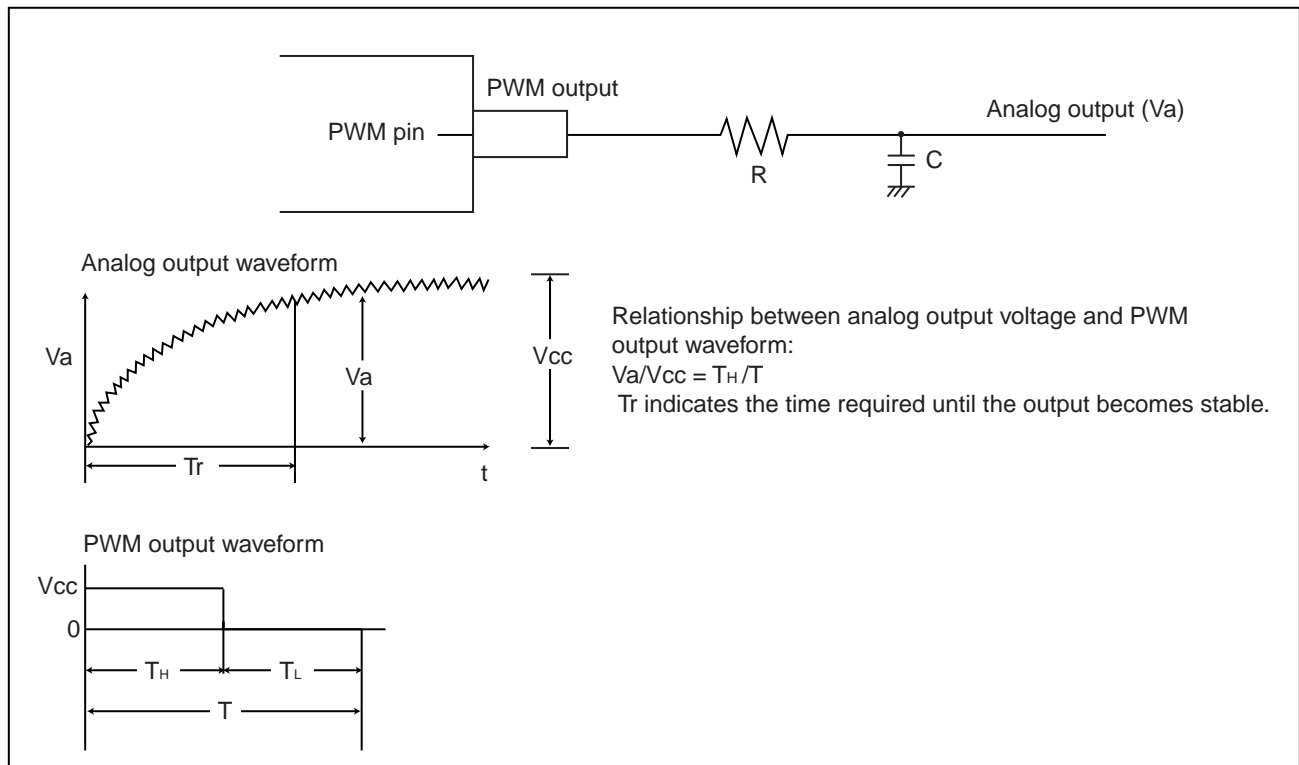
7.2 Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

The 2-channel 8-bit PWM timer consists of two 8-bit PWM timers (CH1, CH2) that are incremented by synchronization with internal count clocks that can be selected from among four types. Each of these 8-bit PWM timers can be selected as an interval timer that can output rectangular waves or a PWM timer with 8-bit or 7-bit resolution.

■ Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

The PWM timer can operate in 8-bit PWM mode, where CH1 and CH2 are used independently; 7-bit PWM mode (high-speed mode); and CH12PWM mode, where a PWM wave is generated by specifying the L width with CH1 and the cycle with CH2. Also, rectangular output from the CH1 timer operating in 8-bit timer mode can be selected as the count clock for the CH2 timer (CK12PWM mode). When a low pass filter is connected to PWM output, the circuit can be used as a D/A converter.

Figure 7.2-1 Example of D/A Converter Configuration with PWM Output and Low Pass Filter



Reference:

Example of calculating the PWM wave (CH12PWM mode)

The PWM wave can be calculated as shown below when the CH1 and CH2 count clock cycles are set to 1 t_{inst} and main clock oscillation (F_{CH}) has been set to 12 MHz and the PWM compare register values have been set to 01_H (COMR1) and 03_H (COMR2).

It is assumed in this example that the fastest clock is selected ($CS1/CS0 = 11_B$, one

instruction cycle = $4/F_{CH}$) as the clock mode ($SCS = 1$) in the system clock control register (SYCC).

$$\begin{aligned} L \text{ width} &= (1 \times 4/F_{CH}) \times (\text{COMR1 value} + 1) \\ &= (4/12 \text{ MHz}) \times (1 + 1) \\ &\text{nearby equal to } 0.67 \mu\text{s} \\ 1 \text{ cycle width} &= (1 \times 4/F_{CH}) \times (\text{COMR2 value} + 1) \\ &= (4/12 \text{ MHz}) \times (3 + 1) \\ &\text{nearby equal to } 1.34 \mu\text{s} \end{aligned}$$

Reference:

During PWM timer operation, interrupt requests are issued only in CH12PWM mode.

■ PWM Timer Function

This function outputs PWM waves by controlling the H width of a cycle or controlling the L width and cycle independently. When a low pass filter is connected to the output, the circuit can also be used as a D/A converter.

When the CH1 and CH2 timers are used independently, 8-bit PWM mode and 7-bit PWM mode (high-speed mode) can be selected for each timer.

○ 8-bit PWM mode

- Since the H width of a cycle can be controlled with resolution to $1/256$, PWM output with a duty ratio from 0 to 99.6% is possible.
- Four PWM wave cycles 2^8 times as long as the four count clock cycles can be selected.

○ 7-bit PWM mode (high-speed mode)

- Since the H width of a cycle can be controlled with resolution to $1/128$, PWM output with a duty ratio from 0 to 99.2% is possible.
- Four PWM wave cycles 2^7 times as long as the four count clock cycles (these lengths are half those in 8-bit PWM mode) can be selected.

○ CK12PWM mode (8-bit PWM, 7-bit PWM)

- 8-bit PWM or 7-bit PWM can be selected for the CH2 timer. Rectangular output from the CH1 timer is always used as the count clock for the CH2 timer.
- For the CH1 timer, only 8-bit timer mode can be used. The PWM wave cycle can be controlled.

○ CH12PWM mode

- The L width of the PWM wave can be controlled for each of the four CH1 count clock cycles, ranging one cycle to 2^8 times the cycle.
- The PWM wave cycle can be controlled for each of the four CH2 count clock cycles, ranging one cycle to 2^8 times the cycle.
- The minimum PWM wave resolution that can be controlled with reduction of the usable duty ratio range is $1/2^{14}$.

7.2 Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

○ Relationship between the PWM wave cycle and mode

Bit functions of the parallel port status register (PSR)

Table 7.2-1 PWM Wave Cycles that can be set with the PWM Timer Function

	Count clock cycle		General mode (CH1 and CH2 are used independently.)		CH12PWM mode	
			8-bit PWM mode	7-bit PWM mode	L width (CH1)	Width of one cycle (CH2)
CH2 in mode other than CK12PWM and CH1	Internal count clock	$1t_{inst}$	2^8t_{inst}	2^7t_{inst}	$1t_{inst}$ to 2^8t_{inst}	$1t_{inst}$ to 2^8t_{inst}
		$8t_{inst}$	$2^{11}t_{inst}$	$2^{10}t_{inst}$	2^3t_{inst} to $2^{11}t_{inst}$	2^3t_{inst} to $2^{11}t_{inst}$
		$16t_{inst}$	$2^{12}t_{inst}$	$2^{11}t_{inst}$	2^4t_{inst} to $1^{12}t_{inst}$	2^4t_{inst} to $2^{12}t_{inst}$
		$64t_{inst}$	$2^{14}t_{inst}$	$2^{13}t_{inst}$	2^6t_{inst} to $2^{14}t_{inst}$	2^6t_{inst} to $2^{14}t_{inst}$
CH2 in CK12PWM mode	CH1 rectangular wave output	$2t_{inst}$ to $2^{15}t_{inst}$	2^9t_{inst} to $2^{23}t_{inst}$	2^8t_{inst} to $2^{22}t_{inst}$	-	

t_{inst} : Instruction cycle (depending on the clock mode)

-: Cannot be set

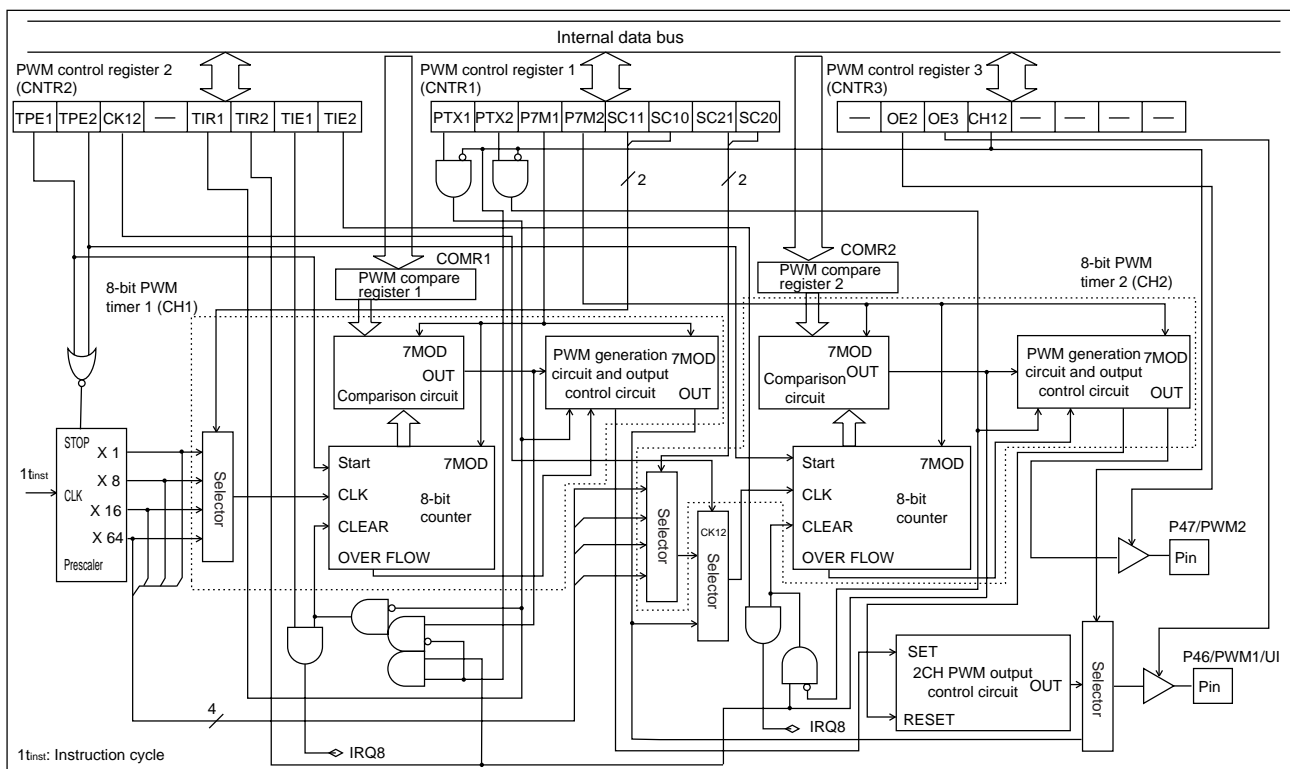
7.3 Configuration of the 2-Channel 8-Bit PWM Timer

The 2-channel 8-bit PWM timer consists of the following four blocks:

- Prescaler
- 8-bit PWM timer 1 (CH1)
- 8-bit PWM timer 2 (CH2)
- PWM compare register1, 2 (COMR1, COMR2)
- PWM control register1, 2, 3 (CNTR1, CNTR2, CNTR3)
- CK12 selector
- CH12PWM output control circuit

■ Block Diagram of the 2-Channel 8-Bit PWM Timer

Figure 7.3-1 Block Diagram of the 2-Channel 8-Bit PWM Timer



■ Prescaler

This circuit divides operating clocks for peripheral circuits.

The circuit is used to output one of the four internal count clocks while either of the counter enable bits (TPE1, TPE2) in the PWM control register (CNTR2) is set to 1.

■ 8-Bit PWM Timer 1 (CH1) and 8-Bit PWM Timer 2 (CH2)

○ Count clock selector

This circuit selects from among four internal count clocks to increment an 8-bit counter value.

○ 8-bit counter

The counter value is incremented by a count clock selected by the count clock selector.

○ Comparison circuit

This circuit has a latch that holds a COMR register value when the 8-bit counter value is 00H. It also compares the 8-bit counter and COMR register value to see if they match.

○ PWM generation circuit and output control circuit

If a match between the 8-bit counter value and COMR register value is detected during interval timer operation, an interrupt request is issued. If output pin control bit 2 or 3 (OE2 or OE3 in CNTR3) is set to 1, the PWM pin output level is inverted by the output control circuit and the 8-bit counter is cleared.

If a match between the 8-bit counter value and COMR register value is detected during PWM timer operation, the PWM pin output level is changed from H to L by the PWM generation circuit. The level is changes back to H if the 8-bit counter overflows.

■ PWM Compare Register 1, 2 (COMR1, COMR2)

These registers store the values that are to be compared to the 8-bit counter values.

■ PWM Control Register 1, 2, 3 (CNTR1, CNTR2, CNTR3)

These registers select operating mode, enable and disable operation, set count clocks, control interrupts, and check the status.

When PWM timer mode is set (PTX = 1), clearing of the 8-bit counter and issuing of an interrupt request are disabled by a signal from the comparison circuit indicating a match.

■ CK12 Selector

This input clock switching circuit switches the clock to be input to the CH2 timer between output from the count clock selector and output from the CH1 timer.

■ CH12PWM Output Control Circuit

This circuit controls the L width (L --> H) and H width (H --> L) of the PWM wave according to the output from the CH1 and CH2 timers.

■ Interrupts Related to the 2-Channel 8-Bit Timer

IRQ8: An interrupt request is issued if interrupt request output has been enabled (TIE1 = 1 in CNTR2) when the counter value matches the COMR1 or COMR2 register value in CH1 interval timer function mode (no interrupt request is issued during ordinary PWM operation).

IRQ8: An interrupt request is issued if interrupt request output has been enabled (TIE1 = 2 in CNTR2) when the counter value matches the COMR1 or COMR2 register value in CH2 interval timer function mode or in CH12PWM mode (no interrupt request is issued during ordinary PWM operation).

149

7.5 Registers of the 2-Channel 8-Bit PWM Timer

This section lists the registers related to the 2-channel 8-bit PWM timer.

■ PWM Control Register 1, 2, 3 (CNTR1, CNTR2, CNTR3)

Figure 7.5-1 Registers Related to 2-Channel 8-Bit PWM Timer

PWM control register 1, 2, 3 (CNTR1, 2, 3)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CNTR1	0 0 2 7 _H	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20	00000000 _B
		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
CNTR2	0 0 2 8 _H	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2	000X0000 _B
		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
CNTR3	0 0 2 9 _H	—	OE2	OE3	CH12	—	—	—	—	X000XXXX _B
		R/W	R/W	R/W						

PWM compare register 1, 2 (COMR1, COMR2)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR1	0 0 2 A _H									XXXXXXXX _B
		W	W	W	W	W	W	W	W	

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR2	0 0 2 B _H									XXXXXXXX _B
		W	W	W	W	W	W	W	W	

R/W :Readable/writable
W :Write-only
- :Unused
X :Undefined

Note:

Because PWM compare registers 1 and 2 (COMR1 and COMR2) are used only for writing (write-only), bit manipulation instructions cannot be used for these registers.

7.5.1 PWM Control Register 1 (CNTR1)

This register selects the operating mode (interval timer operation or PWM timer operation), switches PWM timer resolution, and selects a count clock for the CH1 and CH2 timers.

■ PWM Control Register 1 (CNTR1)

Figure 7.5-2 PWM Control Register 1 (CNTR1)

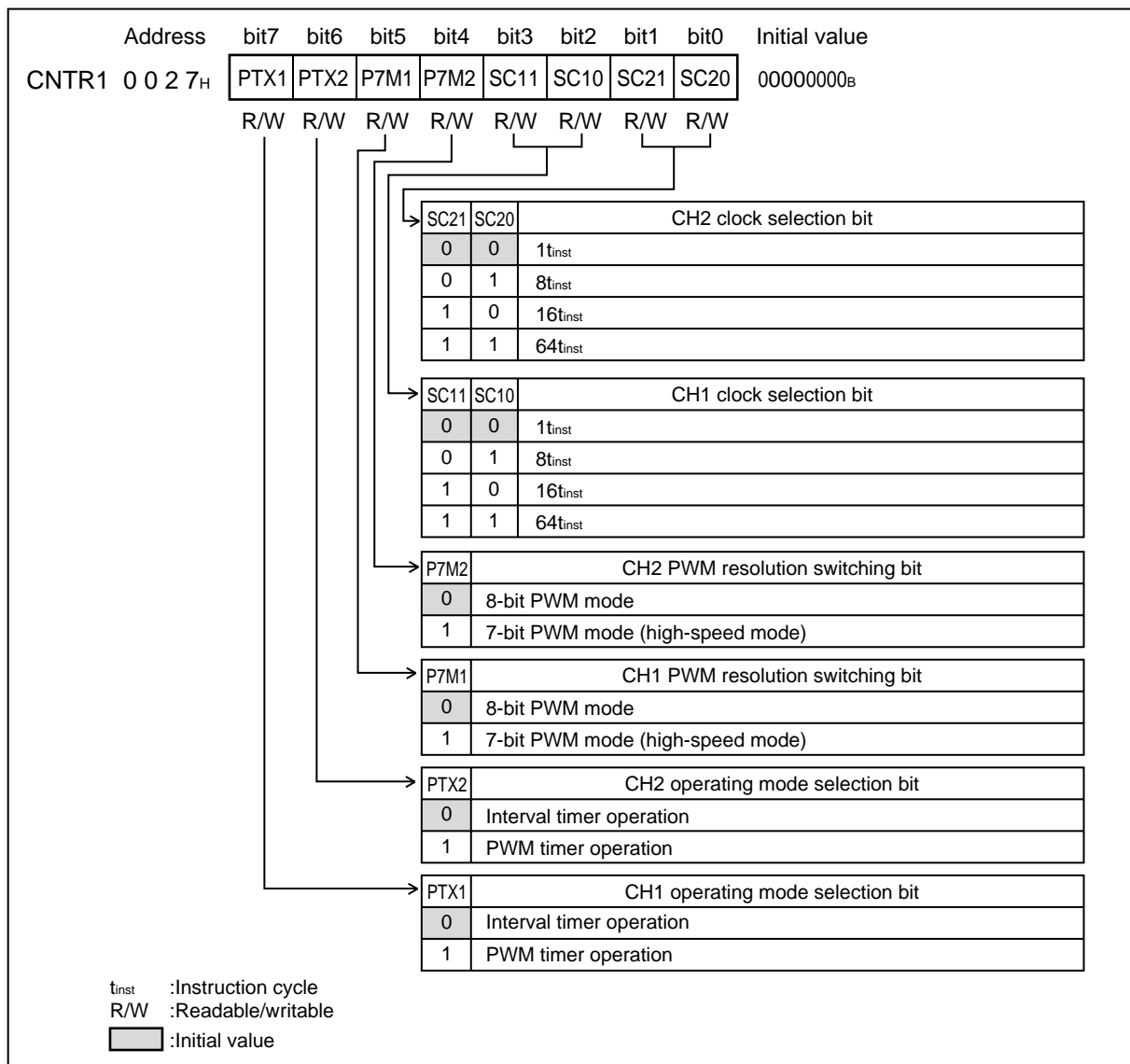


Table 7.5-1 Functions of PWM Control Register 1 (CNTR1) Bits

Bit		Function
bit7	PTX1: CH1 operating mode selection bit	<ul style="list-style-type: none"> This bit selects between interval timer operation and PWM timer operation for the CH1 timer. Interval timer operation is selected when this bit is set to 0. PWM timer operation is selected when this bit is set to 1. <p>Note: Clear the counter stop bit (TPE1 = 0), interrupt disable bit (TIE1 = 0), and interrupt request flag bit (TIR1 = 0) before writing a value to this bit. This bit is invalid in CH12PWM mode (CH12 = 1 in CNTR3).</p>
bit6	PTX2: CH2 operating mode selection bit	<ul style="list-style-type: none"> This bit selects between interval timer operation and PWM timer operation for the CH2 timer. Interval timer operation is selected when this bit is set to 0. PWM timer operation is selected when this bit is set to 1. <p>Note: Clear the counter stop bit (TPE2 = 0), interrupt disable bit (TIE2 = 0), and interrupt request flag bit (TIR2 = 0) before writing a value to this bit. This bit is invalid in CH12PWM mode (CH12 = 1 in CNTR3).</p>
bit5	P7M1: CH1 PWM resolution switching bit	<ul style="list-style-type: none"> This bit switches between 8-bit PWM mode and 7-bit PWM mode (high-speed mode) during PWM timer operation of CH1. 8-bit PWM mode is selected when this bit is set to 0 and 7-bit PWM mode is selected when this bit is set to 1. <p><Note> Do not write 1 to this bit during interval timer operation.</p>
bit4	P7M2: CH2 PWM resolution switching bit	<ul style="list-style-type: none"> This bit switches between 8-bit PWM mode and 7-bit PWM mode (high-speed mode) during PWM timer operation of CH2. 8-bit PWM mode is selected when this bit is set to 0. 7-bit PWM mode is selected when this bit is set to 1. <p>Note: Do not write 1 to this bit during interval timer operation.</p>
bit3 bit2	SC11, SC10: CH1 clock selection bit	<ul style="list-style-type: none"> This bit selects a count clock for the CH1 interval timer/PWM timer function. There are four selectable internal count clocks. <p>Note: Do not manipulate this bit while the CH1 counter is operating (TPE1 = 1).</p>
bit1 bit0	SC21, SC20: CH2 clock selection bit	<ul style="list-style-type: none"> This bit selects a count clock for the CH2 interval timer/PWM timer function. There are four selectable internal count clocks. <p>Note: Do not manipulate this bit while the CH2 counter is operating (TPE2 = 1).</p>

7.5.2 PWM Control Register 2 (CNTR2)

This register enables and disables 8-bit PWM timer operation, selects CK12 mode, controls interrupts, and checks the status of the CH1 and Ch2 timers.

■ PWM Control Register 2 (CNTR2)

Figure 7.5-3 PWM Control Register 2 (CNTR2)

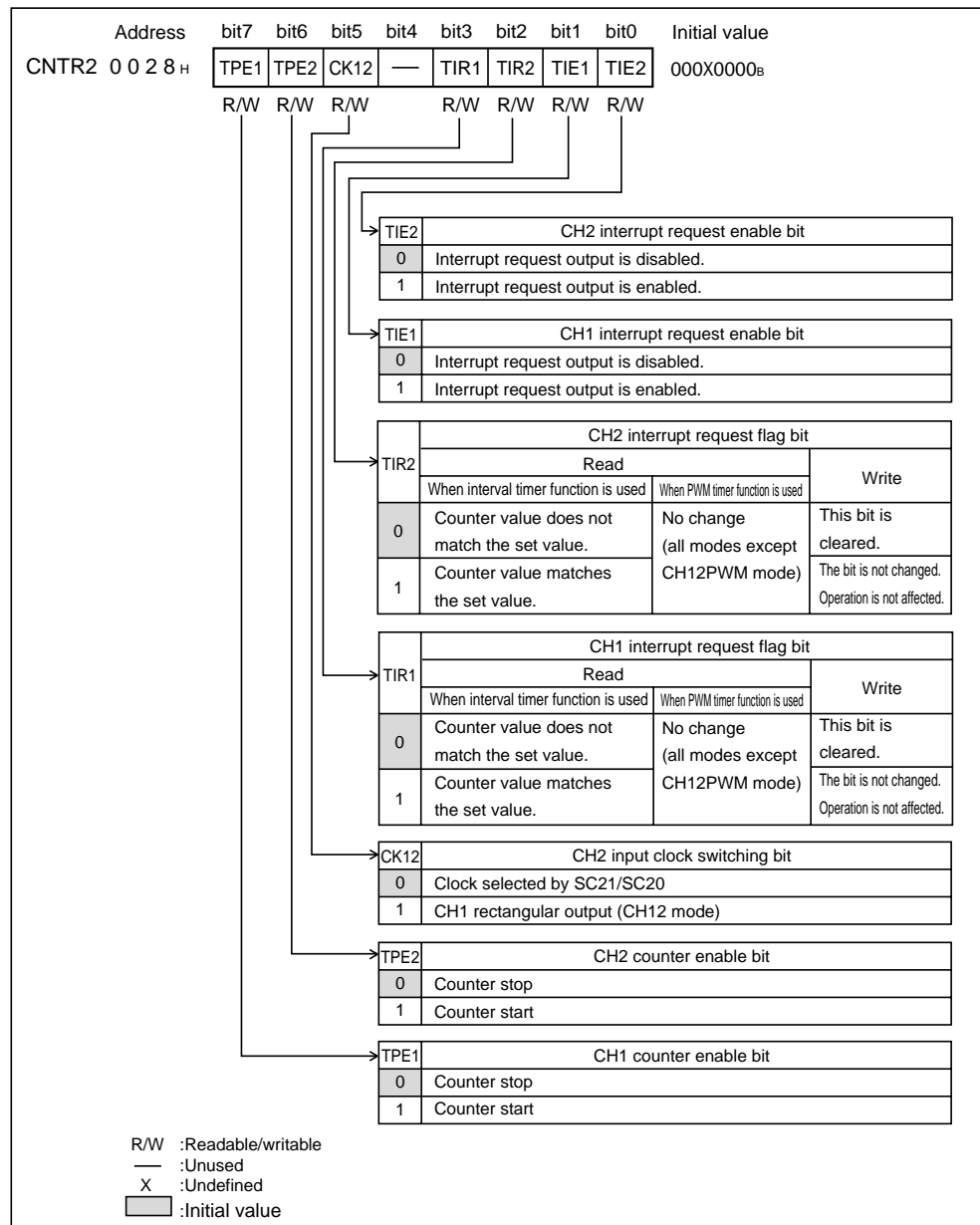


Table 7.5-2 Functions of PWM Control Register 2 (CNTR2) Bits

Bit		Function
bit7	TPE1: CH1 counter enable bit	<ul style="list-style-type: none"> This bit starts and stops CH1 interval timer operation or PWM timer operation. Counting starts when this bit is set to 1 and the counter value is cleared to 00_H. Counting stops when this bit is set to 0.
bit6	TPE2: CH2 counter enable bit	<ul style="list-style-type: none"> This bit starts and stops CH2 interval timer operation or PWM timer operation. Counting starts when this bit is set to 1 and the counter value is cleared to 00_H. Counting stops when this bit is set to 0.
bit5	CK12: CH2 input clock switching bit	<ul style="list-style-type: none"> This bit switches the input clock to CH2. The clock selected by SC21/SC20 bit is input to CH2 when this bit is set to 0. When this bit is set to 1, CH1, rectangular output is input to CH2 and CK12 mode is set regardless of the SC21/SC20 bit value. <p>Note: Do not write 1 to this bit in CH12PWM mode (CH12 = 1 in CNTR3). Also do not write 1 while CH1 is operating as the PWM timer (PTX1 = 1 in CNTR1).</p>
bit4	Unused	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit3	TIR1: CH1 interrupt request flag bit	<ul style="list-style-type: none"> This bit is set to 1 when the count matches the COMR1 register value while the CH1 interval timer function is used. An interrupt request is issued to the CPU when this bit and the CH1 interrupt request enable bit (TIE1) are set to 1. No interrupt requests are issued when the PWM timer function is being used in a mode other than CH12PWM mode. Writing 0 to this bit clears it. Writing 1 to this bit has no effect.
bit2	TIR2: CH2 interrupt request flag bit	<ul style="list-style-type: none"> This bit is set to 1 when the count matches the COMR2 register value when the CH2 interval timer function is being used or CH12PWM mode is set. An interrupt request is issued to the CPU when this bit and CH2 interrupt request enable bit (TIE2) are set to 1. No interrupt requests are issued when the PWM timer function is being used in a mode other than CH12PWM mode. Writing 0 to this bit clears it. Writing 1 to this bit has no effect.
bit1	TIE1: CH1 interrupt request enable bit	<ul style="list-style-type: none"> This bit enables or disables interrupt requests output to the CPU for CH1. An interrupt request is issued when this bit and the TIR1 bit are set to 1. <p>Note: Disable interrupt request output (TIE1 = 0) when CH12PWM mode is used (CH12 = 1 in CNTR3).</p>
bit0	TIE2: CH2 interrupt request enable bit	<ul style="list-style-type: none"> This bit enables or disables interrupt request output to the CPU for CH2. An interrupt request is issued when this bit and the TIR2 bit are set to 1.

7.5.3 PWM Control Register 3 (CNTR3)

This register selects CH12PWM mode and controls the output pins for the CH1 and CH2 timers.

■ PWM Control Register 3 (CNTR3)

Figure 7.5-4 PWM Control Register 3 (CNTR3)

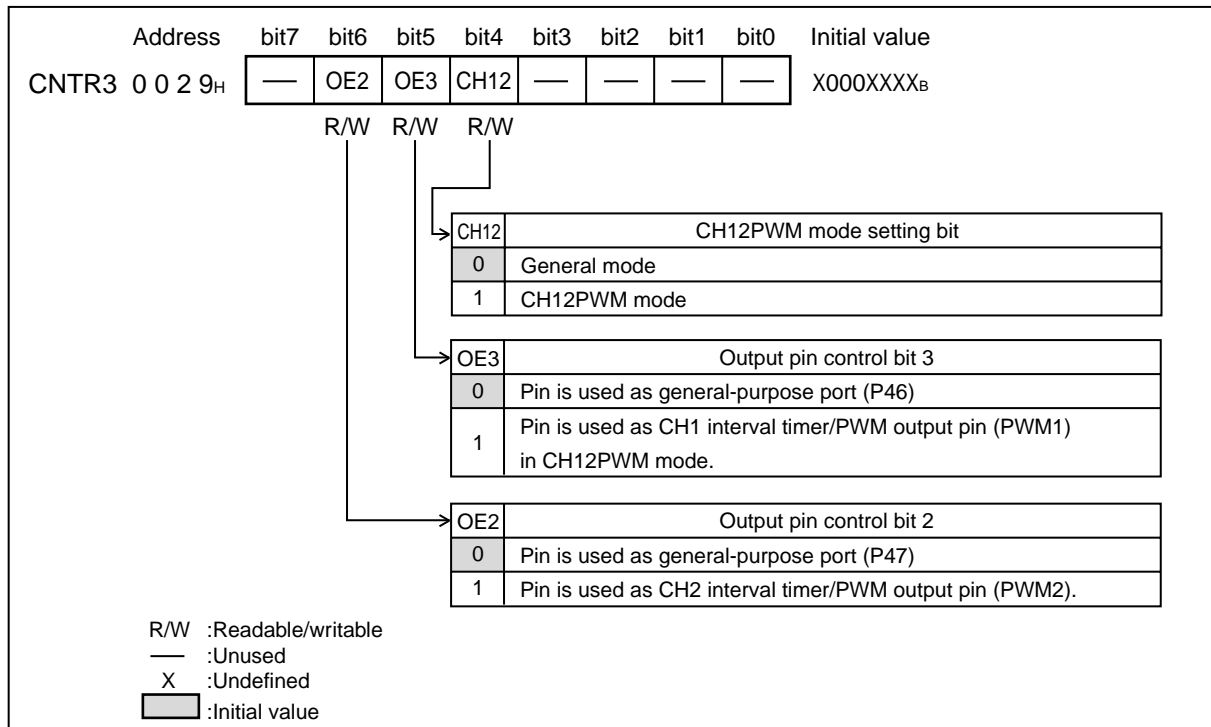


Table 7.5-3 Functions of PWM Control Register 3 (CNTR3) Bits

Bit		Function
bit7	Unused	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.
bit6	OE2: Output pin control bit 2	<ul style="list-style-type: none"> This bit switches the P47/PWM2 pin function between general-purpose port and dedicated pin. The P47/PWM2 pin is used as a general-purpose port (P47) when this bit is set to 0, and is used as a dedicated pin (PWM2) when this bit is set to 1. The PWM2 pin outputs rectangular waves from CH2 when the interval timer function is used, and PWM waves when the PWM timer function is used.
bit5	OE3: Output pin control bit 3	<ul style="list-style-type: none"> This bit switches the P46/PWM1/UI pin function between general-purpose port and dedicated pin. The P47/PWM2 pin is used as a general-purpose port (P46) when this bit is set to 0, and is used as a dedicated pin (PWM1) when this bit is set to 1. The PWM1 pin outputs rectangular waves from CH1 when the interval timer function is used, and PWM waves when the PWM timer function is used. It outputs PWM waves in CH12PWM mode.
bit4	CH12: PWM mode setting bit	<ul style="list-style-type: none"> This bit switches between general mode and CH12PWM mode. CH1 and CH2 can operate independently when this bit is set to 0. CH12PWM mode in which the L width is specified by CH1 and the cycle is specified by CH2 is set when this bit is set to 1. The operating mode selection bits (PTX1, PTX2 in CNTR1) are invalid in CH12PWM mode. <p>Note: Do not write 1 to this bit in CK12 mode (CK12 = 1 in CNTR2). Do not change this bit while the CH1 or CH2 counter is operating (TIE1 = 1 or TIE2 = 1 in CNTR2).</p>
bit3 bit2 bit1 bit0	Unused	<ul style="list-style-type: none"> The read value is undefined. Writing to this bit has no effect on operation.

7.5.4 PWM Compare Register 1 (COMR1)

This register is a CH1 timer data register. The register indicates the interval when the interval timer function is used, and indicates the pulse H width when the general PWM timer function is used. In CH12PWM mode, it indicates the pulse L width.

■ PWM Compare Register 1 (COMR1)

Figure 7.5-5 "PWM Compare Register 1 (COMR1)" shows the bit configuration of this register.

Since this register can only be written to (write-only), bit manipulation instructions cannot be used on it.

Figure 7.5-5 PWM Compare Register 1 (COMR1)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR1 0 0 2 A _H	*								XXXXXXXX _B
	W	W	W	W	W	W	W	W	

W :Write-only
 X :Undefined
 * :Invalid in seven-bit PWM mode (high-speed mode)

○ Function during interval timer operation (8-bit timer mode and CK12 mode)

This register stores the value to be compared to the counter value and specifies an interval (rectangular output frequency).

The counter is cleared and the interrupt request flag bit is set to 1 (TIR1 = 1 in CNTR2) if a value written to this register matches the counter value.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle (after matching is detected).

Reference:

The value to be written to this register in 8-bit timer operation and CK12 mode operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

$$\text{COMR1 register value} = \text{interval} / (\text{count clock cycle} \times \text{instruction cycle}) - 1$$

○ Function during PWM timer operation (8-bit PWM mode and 7-bit PWM mode)

This register stores the value to be compared to the counter value and specifies the pulse H width.

H level output from the PWM1 pin continues until a value written to this register matches the counter value. L level output starts when there is a match and continues until the counter overflows.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle (after overflow).

Reference:

The value to be written to this register and the cycle to be set during PWM timer operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

- 8-bit PWM mode
 - $\text{COMR1 register value} = \text{duty ratio} \times 256$
 - $\text{PWM wave cycle} = \text{count clock cycle} \times \text{instruction cycle} \times 256$
- 7-bit PWM mode
 - $\text{COMR1 register value} = \text{duty ratio} \times 128$
 - $\text{PWM wave cycle} = \text{count clock cycle} \times \text{instruction cycle} \times 128$

○ Function in CH12PWM mode operation

This register stores the value to be compared to the counter value and specifies a pulse L width.

L level output from the PWM1 pin continues until a value written to this register matches the counter value. The output is switched to the H level when there is a match.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle.

Reference:

The value to be written to this register in CH12PWM mode operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

$$\text{COMR1 register value} = \text{PWM wave L width period} / (\text{count clock cycle} \times \text{instruction cycle}) - 1$$

7.5.5 PWM compare register 2 (COMR2)

This register is a CH2 timer data register. The register indicates the interval when the interval timer function is used, and indicates the pulse H width when the general PWM timer function is used. In CH12PWM mode, it indicates the PWM wave cycle.

■ PWM Compare Register 2 (COMR2)

Figure 7.5-6 "PWM Compare Register 2 (COMR2)" shows the bit configuration of this register.

Since this register can only be written to (write-only), bit manipulation instructions cannot be used on it.

Figure 7.5-6 PWM Compare Register 2 (COMR2)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
COMR12 0 0 2 B _H	*								XXXXXXXX _B
	W	W	W	W	W	W	W	W	

W :Write-only
 X :Undefined
 * :Invalid in seven-bit PWM mode (high-speed mode)

○ Function during interval timer operation (8-bit timer mode and CK12 mode)

This register stores the value to be compared to the counter value and specifies an interval.

The counter is cleared and the interrupt request flag bit is set to 1 (TIR2 = 1 in CNTR2) if a value written to this register matches the counter value.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle (after a match is detected).

Reference:

The value to be written to this register during interval timer operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

- 8-bit timer mode
 - $\text{COMR2 register value} = \text{interval} / (\text{count clock cycle} \times \text{instruction cycle}) - 1$
- CK12 mode
 - $\text{COMR2 register value} = \text{interval} / \text{CH1 rectangular output} - 1$

○ Function during PWM timer operation (8-bit PWM mode and 7-bit PWM mode)

This register stores the value to be compared to the counter value and specifies a pulse H width.

H level output from the PWM2 pin continues until a value written to this register matches the counter value. L level output starts when there is a match and continues until the counter overflows.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle (after overflow).

Reference:

The value to be written to this register and cycle to be set during PWM timer operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

- 8-bit PWM mode
 - $\text{COMR2 register value} = \text{duty ratio} \times 256$
 - $\text{PWM wave cycle} = \text{count clock cycle} \times \text{instruction cycle} \times 256$
- 7-bit PWM mode
 - $\text{COMR2 register value} = \text{Duty ratio} \times 128$
 - $\text{PWM wave cycle} = \text{count clock cycle} \times \text{instruction cycle} \times 128$

○ Function in CH12PWM mode operation

This register stores the value to be compared to the counter value and specifies a PWM wave cycle.

The CH1 and CH2 counters are cleared and the interrupt request flag bit is set to 1 ($\text{TIR2} = 1$ in CNTR2) if a value written to this register matches the counter value.

The output from the PWM1 pin is switched to the L level.

When a value is written to this register while the counter is operating, the value takes effect in the next cycle.

Reference:

The value to be written to this register in CH12PWM mode operation can be calculated using the expression shown below. Note that the instruction cycle varies depending on the clock mode and the gear function.

$\text{COMR2 register value} = \text{period of one PWM wave cycle} / (\text{count clock cycle} \times \text{instruction cycle}) - 1$

7.6 Interrupts of the 2-Channel 8-Bit PWM Timer

An interrupt is generated if the counter value matches a PWM compare register value when the interval timer function is used. No interrupt requests are issued when the PWM timer function is used in a mode other than CH12PWM mode.

■ Interrupts

If the counter value is incremented from 00_H to a PWM compare register (COMR) value by a selected count clock, the corresponding interrupt request flag bit (TIR1 or TIR2 in CNTR2) is set to 1.

If the interrupt request enable flag has been set (TIE1 = 1/TIE2 = 1 in CNTR2) to enable interrupt request output, an interrupt request (IRQ8) is issued to the CPU. Clear the interrupt request by using the interrupt handling routine to write 0 to the TIR bit.

The TIR1 and TIR2 bits are set to 1 when they match the CH1 or CH2 counter value regardless of the TIE1 and TIE2 bit values.

Note:

Disable interrupts for the CH1 timer (TIE1 = 0 in CNTR2) in CH12PWM mode. For the CH2 timer, they can be used the same way they are used for the interval timer function.

Reference:

If the counter value and register value match at the same time the counter has stopped (TPE1 = 0, TPE2 = 0 in CNTR2), the TIR bit is not set to 1.

If the TIE bit is changed from 0 --> 1 (from disabled to enabled) while the TIR bit is set to 1, an interrupt request is issued immediately.

■ Register and Vector Table Related to Interrupts

Table 7.6-1 Register and Vector Table Related to Interrupts

Interrupt	Interrupt level setting register			Vector table address	
	Register	Setting bit		Higher	Lower
IRQ8	ILR3(007E _H)	L81(bit1)	L80(bit0)	FFEA _H	FFEB _H

Reference:

See Section 3.4.2 "Processing During Interrupt Operation" for interrupt operation.

7.7 Operations of the Interval Timer Function

This section explains the operation of the interval timer function in 8-bit timer mode and CK12 mode.

■ Operation of the Interval Timer Function

Operation of the CH1 and CH2 timers as an interval timer in 8-bit timer mode or CK12 mode requires the setting shown in Figure 7.7-1 "Setting for Interval Timer Function".

Figure 7.7-1 Setting for Interval Timer Function

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1...	0	×	0	×	⊙	⊙	×	×
CH2...	×	0	×	0	×	×	⊙	⊙
CK12...	0	0	0	0	⊙	⊙	×	×

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
CH1...	1	×	0		⊙	×	⊙	×
CH2...	×	1	0		×	⊙	×	⊙
CK12...	1	1	1		⊙	⊙	⊙	⊙

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1...		×	⊙	0				
CH2...		⊙	×	0				
CK12...		⊙	⊙	0				

COMR1	CH1 interval (compare value) is set.
COMR2	CH2 interval (compare value) is set.

⊙ :Used
1 :Set to 1
0 :Set to 0
× :Unused (set to 0)

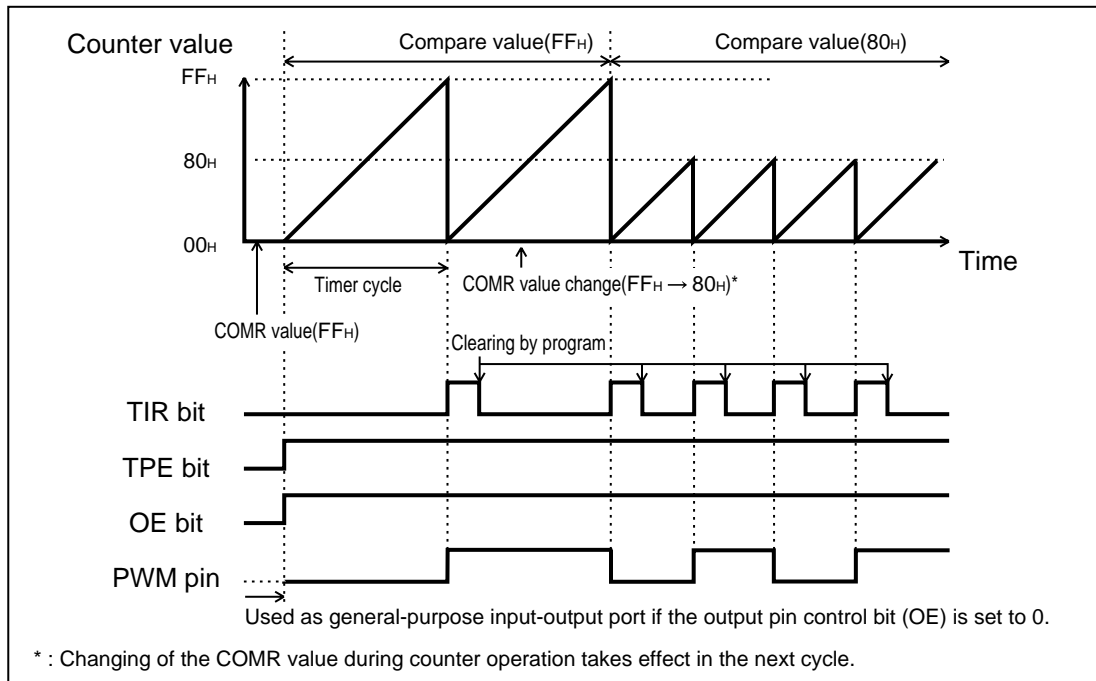
CH1 :CH1 in eight-bit timer mode
CH2 :CH2 in eight-bit timer mode
CK12 :CK12 mode

The counter value is incremented from 00_H on the leading edge of the selected count clock after the counter is activated. If the counter value matches the COMR register value (compare value), the PWM pin level is inverted on the next count clock leading edge. The counter is also cleared, the interrupt request flag bit is set (TIR1 = 1/TIR2 = 1 in CNTR2), and the counter value is incremented from 00_H again.

In CK12 mode, rectangular output from the CH1 timer is used as input clock to the CH2 timer.

Figure 7.7-2 "Operation of the Interval Timer Function" shows the operation.

Figure 7.7-2 Operation of the Interval Timer Function



Reference:

- When the COMR register is reset to 00_H, the PWM pin output is inverted in the selected count clock cycle.
- When the interval timer function is used and the counter has stopped (TPE1 = 0 or TPE2 = 0 in CNTR2), the PWM pin output level is L.

Note:

Do not change the corresponding count clock cycle (SC11/SC10 or SC21/SC20 in CNTR1) while the interval timer function is operating (TPE1 = 1 or TPE2 = 1 in CNTR2). The SC21 and SC20 bits are invalid in CK12 mode.

7.8 Operations of the 8-Bit PWM Mode

This section explains operation in 8-bit PWM mode.

■ Operation of the PWM Timer Function

Operation of the CH1 and CH2 timers as a PWM timer in 8-bit PWM mode requires the setting shown in Figure 7.8-1 "Setting for 8-bit PWM Mode".

Figure 7.8-1 Setting for 8-bit PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1...	1	×	0	×	⊙	⊙	×	×
CH2...	×	1	×	0	×	×	⊙	⊙
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
CH1...	1	×	0		×	×	×	×
CH2...	×	1	0		×	×	×	×
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1...		×	1	0				
CH2...		1	×	0				
COMR1	CH1 pulse H width (compare value) is set.							
COMR2	CH2 pulse H width (compare value) is set.							

⊙ :Used
1 :Set to 1
0 :Set to 0
× :Unused (set to 0)

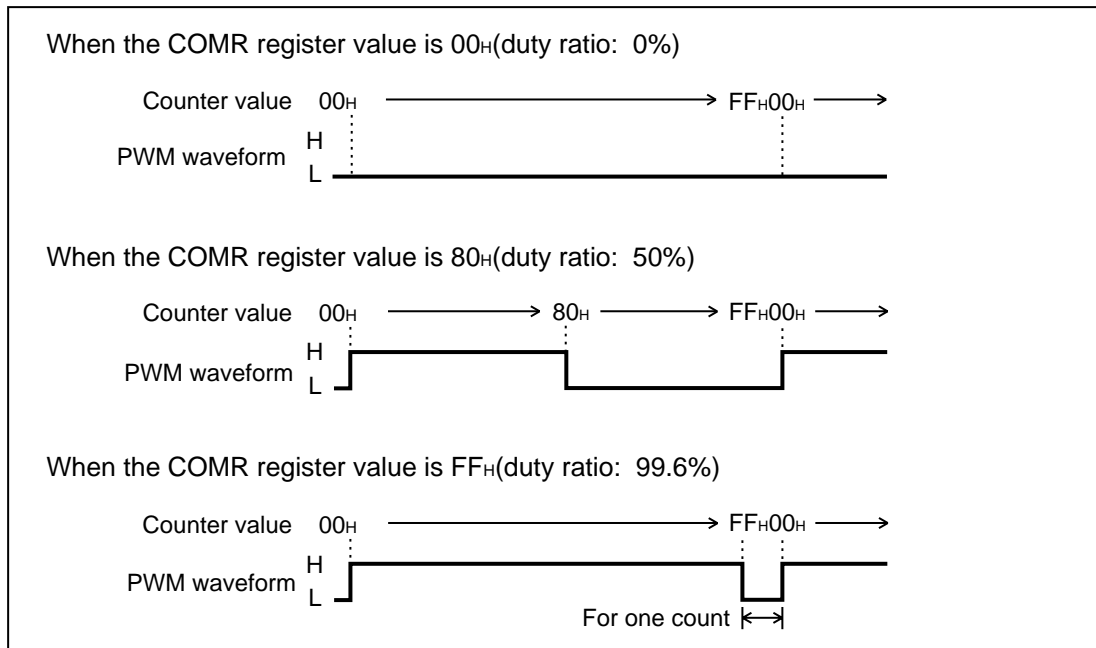
CH1 :CH1 in eight-bit timer mode
CH2 :CH2 in eight-bit timer mode

The counter value is incremented from 00_H on the leading edge of the selected count clock after the counter is activated. The PWM pin output (PWM waveform) level is H until the counter value matches the COMR register value. It changes to L when there is a match and remains L until the counter overflows (FF_H --> 00_H).

If the CK12 bit is set to 1, the CH1 timer operates in 8-bit timer mode, the CH2 timer operates in PWM mode, and CK12PWM mode is set.

Figure 7.8-2 "Example of PWM Waveform in 8-bit PWM Mode" shows a PWM waveform output to the PWM pin.

Figure 7.8-2 Example of PWM Waveform in 8-bit PWM Mode



Reference:

When the PWM timer function is used and the counter (TPE1 = 0 or TPE2 = 0 in CNTR2) has stopped, the PWM pin output level is not changed.

Note:

- Do not change the corresponding count clock cycle (SC11/SC10 or SC21/SC20 in CNTR1) while the PWM timer function is operating (TPE1 = 1 or TPE2 = 1 in CNTR2).
- CK12 mode (CK12 = 1 in CNTR2) cannot be set when the CH1 is during the PWM timer operation.

7.9 Operations of the 7-Bit PWM Mode

This section explains operation in 7-bit PWM mode (high-speed mode).

■ Operation of the High-speed PWM Timer Function

Operation of the CH1 and CH2 timers as a PWM timer in 7-bit PWM mode requires the setting shown in Figure 7.9-1 "Setting for 7-bit PWM Mode".

Figure 7.9-1 Setting for 7-bit PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
CH1...	1	×	1	×	⊙	⊙	×	×
CH2...	×	1	×	1	×	×	⊙	⊙
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CH12	—	TIR1	TIR2	TIE1	TIE2
CH1...	1	×	0		×	×	×	×
CH2...	×	1	0		×	×	×	×
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
CH1...		×	1	0				
CH2...		1	×	0				
	bit7							
COMR1	—	CH1 pulse H width (compare value) is set.						
	bit7							
COMR2	—	CH2 pulse H width (compare value) is set.						

⊙ :Used
 1 :Set to 1
 0 :Set to 0
 × :Unused (set to 0)

CH1 :CH1 in seven-bit timer mode
 CH2 :CH2 in seven-bit timer mode

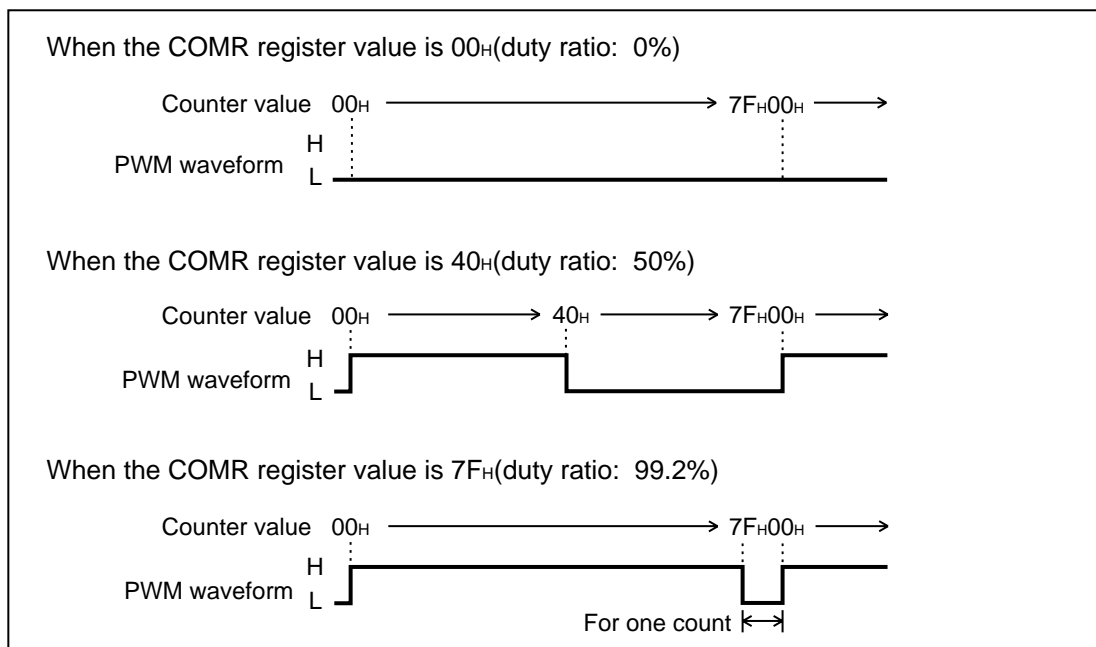
The counter value is incremented from 00H on the leading edge of the selected count clock after the counter is activated. The PWM pin output (PWM waveform) level is H until the counter value matches the COMR register value. It changes to L when there is a match and remains L until the counter overflows (7FH --> 00H).

Operation in 7-bit PWM mode is quicker than that in 8-bit PWM mode because the number of bits in the counter in 7-bit PWM mode is one less than in 8-bit PWM mode. In other words, the PWM frequency in 7-bit PWM mode is twice the frequency in 8-bit PWM mode (the cycle is half that in 8-bit PWM mode).

If the CK12 bit is set to 1, the CH1 timer operates in 8-bit timer mode, the CH2 timer operates in PWM mode, and CK12PWM mode is set.

Figure 7.9-2 "Example of PWM Waveform In 7-bit PWM Mode" shows PWM waveform output to the PWM pin.

Figure 7.9-2 Example of PWM Waveform In 7-bit PWM Mode



Reference:

When the PWM timer function is used and the counter has stopped (TPE1 = 0 or TPE2 = 0 in CNTR2), the PWM pin output level is not changed.

Note:

- Do not change the corresponding count clock cycle (SC11/SC10 or SC21/SC20 in CNTR1) while the PWM timer function is operating (TPE1 = 1 or TPE2 = 1 in CNTR2).
- CK12 mode (CK12 = 1 in CNTR2) cannot be set when the CH1 is during the PWM timer operation.

7.10 Operations of the CH12PWM Mode

This section explains operation in CH12PWM mode.

■ CH12PWM Mode Operation

Operation of the CH1 and CH2 timers as a PWM timer in CH12PWM mode requires the setting shown in Figure 7.10-1 "Setting for CH12PWM Mode".

Figure 7.10-1 Setting for CH12PWM Mode

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR1	PTX1	PTX2	P7M1	P7M2	SC11	SC10	SC21	SC20
	×	×	0	0	⊙	⊙	⊙	⊙
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR2	TPE1	TPE2	CK12	—	TIR1	TIR2	TIE1	TIE2
	1	1	0		×	⊙	0	⊙
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
CNTR3	—	OE2	OE3	CH12	—	—	—	—
		0	1	1				
COMR1	PWM wave L width (compare value) is set.							
COMR2	One PWM wave cycle (compare value) is set.							

⊙ :Used
1 :Set to 1
0 :Set to 0
× :Unused (set to 0)

The CH1 and CH2 counters are incremented from 00H on the leading edge of the selected count clocks. The PWM pin output (PWM waveform) level is L until the CH1 counter value matches the COMR1 register value. It changes to H when there is a match. CH1 and CH2 are cleared to allow incrementing from 00H again if the CH2 counter value matches the COMR2 register value. The PWM pin output is also changed to L, and the interrupt request flag bit is set (TIR2 = 1 in CNTR2).

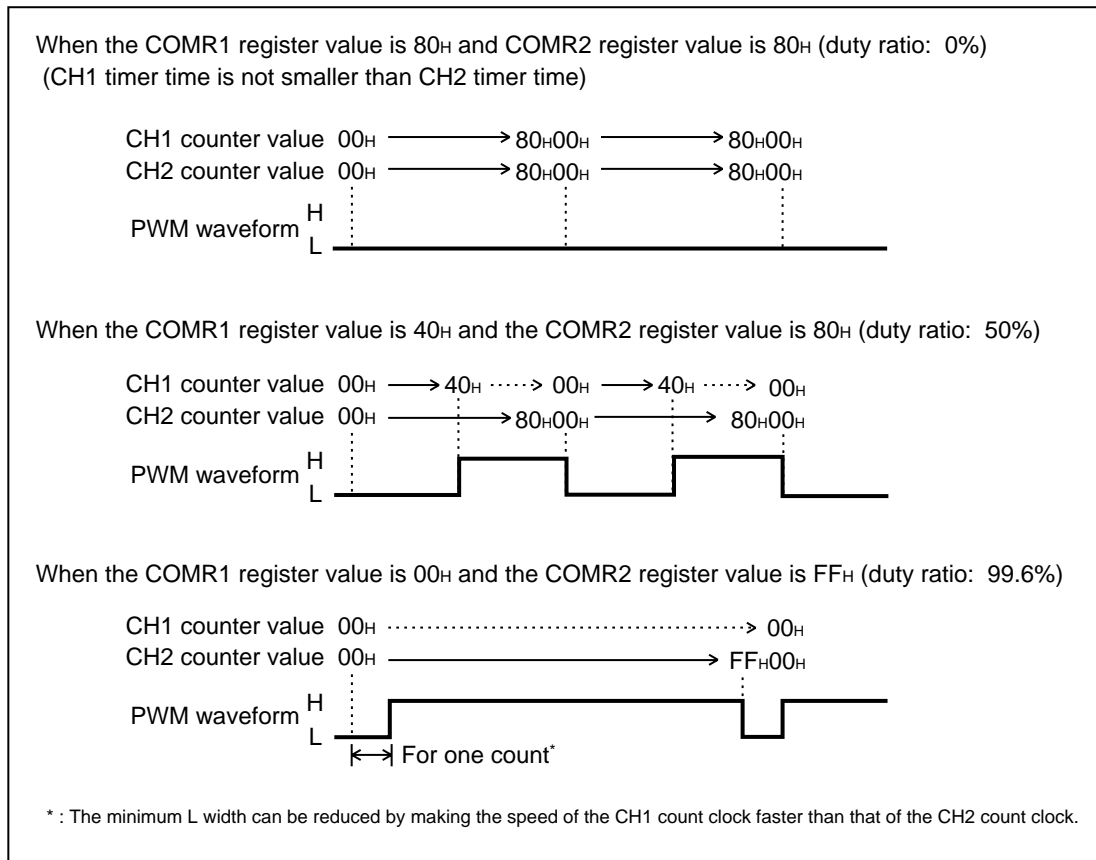
Simultaneously activate the CH1 and CH2 counters (TPE1 = 1 and TPE2 = 1 in CNTR2). If they are activated separately, the L width of the first PWM wave cycle or the cycle is affected.

The PWM wave is not generated if the CH1 timer time that is to be the PWM wave L width is larger than the CH2 timer time that is to be the PWM wave cycle.

Figure 7.10-2 "PWM Waveform (PWM Pin) Output Example" shows examples of the PWM waveforms output to the PWM pin when the same count clock cycle is specified with the CH1 and CH2 timers.

When the PWM timer function is used and the counter has stopped (TPE1 = 0 or TPE2 = 0 in CNTR2), the PWM pin output level is not changed.

Figure 7.10-2 PWM Waveform (PWM Pin) Output Example



Note:

Do not change the corresponding count clock cycles (SC11/SC10 and SC21/SC20 in CNTR1) while the PWM timer function is operating (TPE1 = 1 and TPE2 = 1 in CNTR2).

CK12 mode (CK12 = 1 in CNTR2) cannot be set in CH12PWM mode. Also disable CH1 interrupt request output (TIE1 = 0 in CNTR2).

7.11 Prescaler Operation of the 2-Channel 8-Bit PWM Timer

This section explains operation of the prescaler.

■ Prescaler Operation

Operation is enabled while either of the counter enable bits (TPE1 and TPE2 in CNTR2) is set to 1.

Consequently, the CH1 and CH2 timers operate the same way, with no difference in the cycle, when the TPE1 and TPE2 bits are set to 1 simultaneously.

If the TPE1 or TPE2 bit is set to 1 while the TPE2 or TPE1 bit is set to 1, counting by the CH1 and CH2 timers starts asynchronously, resulting in a difference in the first cycle of one count clock cycle or less.

Figure 7.11-1 "Prescaler Operation" and Figure 7.11-2 "Prescaler Output" show the prescaler operation.

Figure 7.11-1 Prescaler Operation

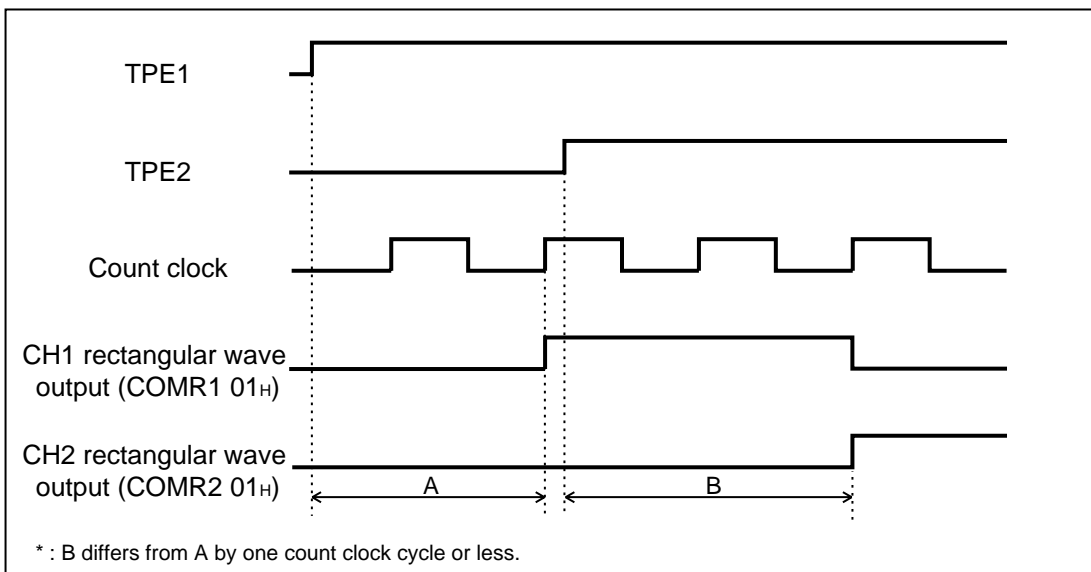
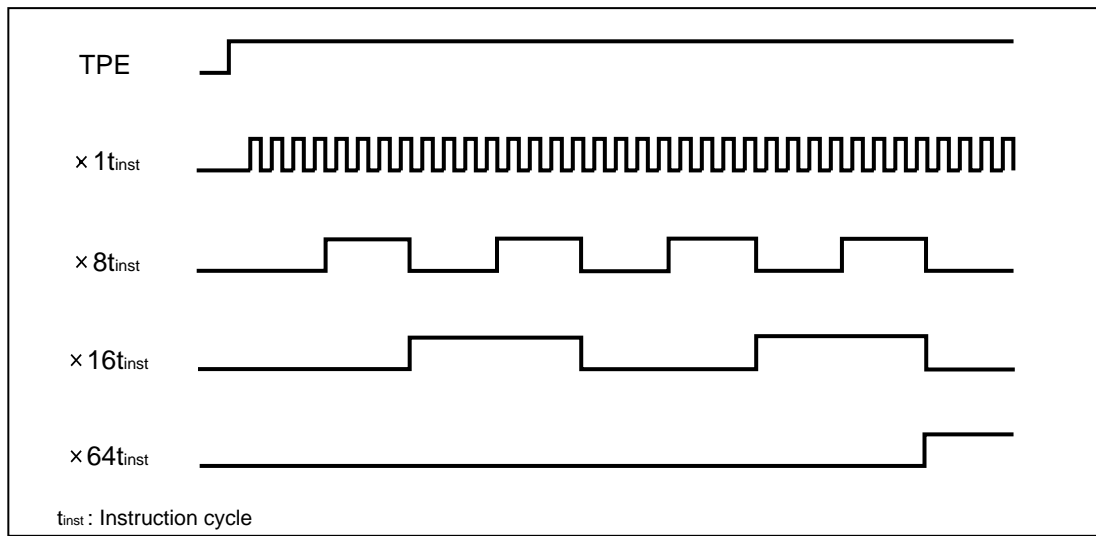


Figure 7.11-2 Prescaler Output



7.12 Operation for Standby Mode and Intermediate Stop

This section explains operation when there is a switch to sleep mode or stop mode or an intermediate stop is requested.

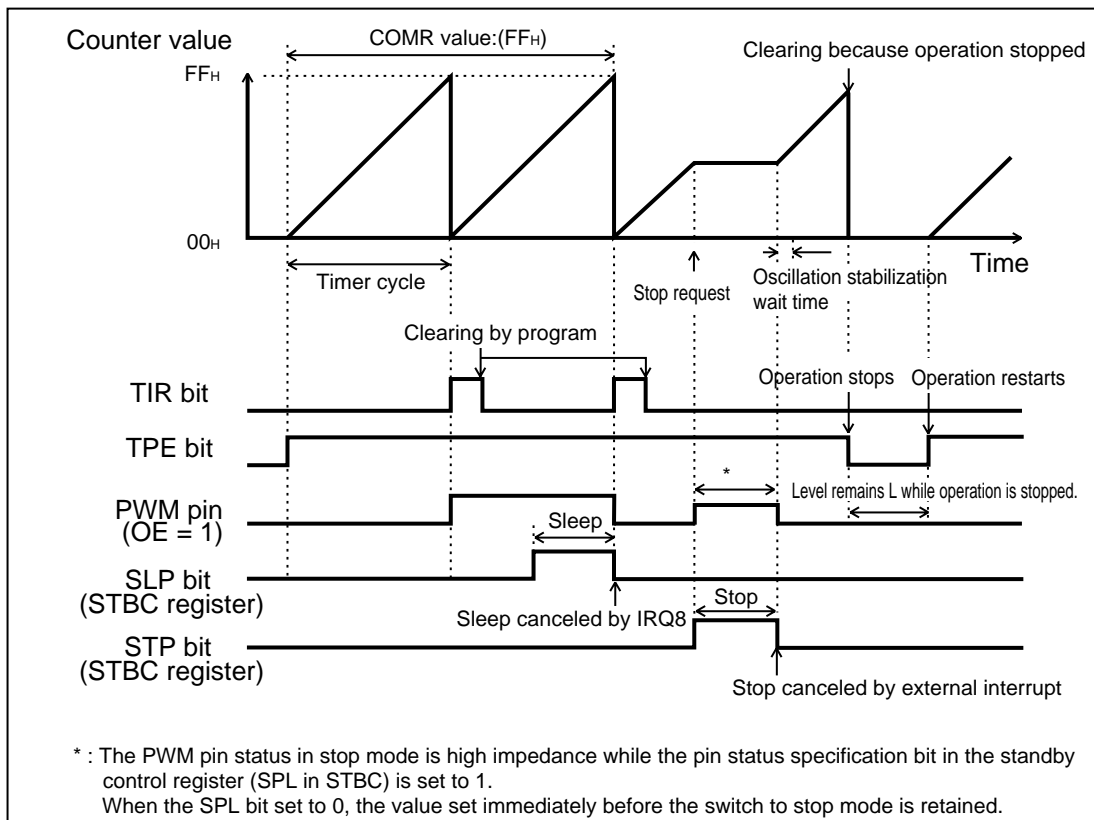
■ Operation for Standby Mode and Intermediate Stop

Figure 7.12-1 "Counter operation for standby mode and intermediate stop (interval function)" and Figure 7.12-2 "Counter Operation for Standby Mode and Intermediate Stop (PWM Timer Function)" show the counter status when there is a switch to sleep mode or stop mode or intermediate stop has been requested during interval timer operation or PWM timer operation.

If the mode switches to stop mode, the counter is stopped and the value retained. If this stop mode is released by an external interrupt, the counter restarts using the retained value. The first interval and PWM cycle therefore differ from the set values. Reinitialize the 8-bit PWM timer after releasing stop mode.

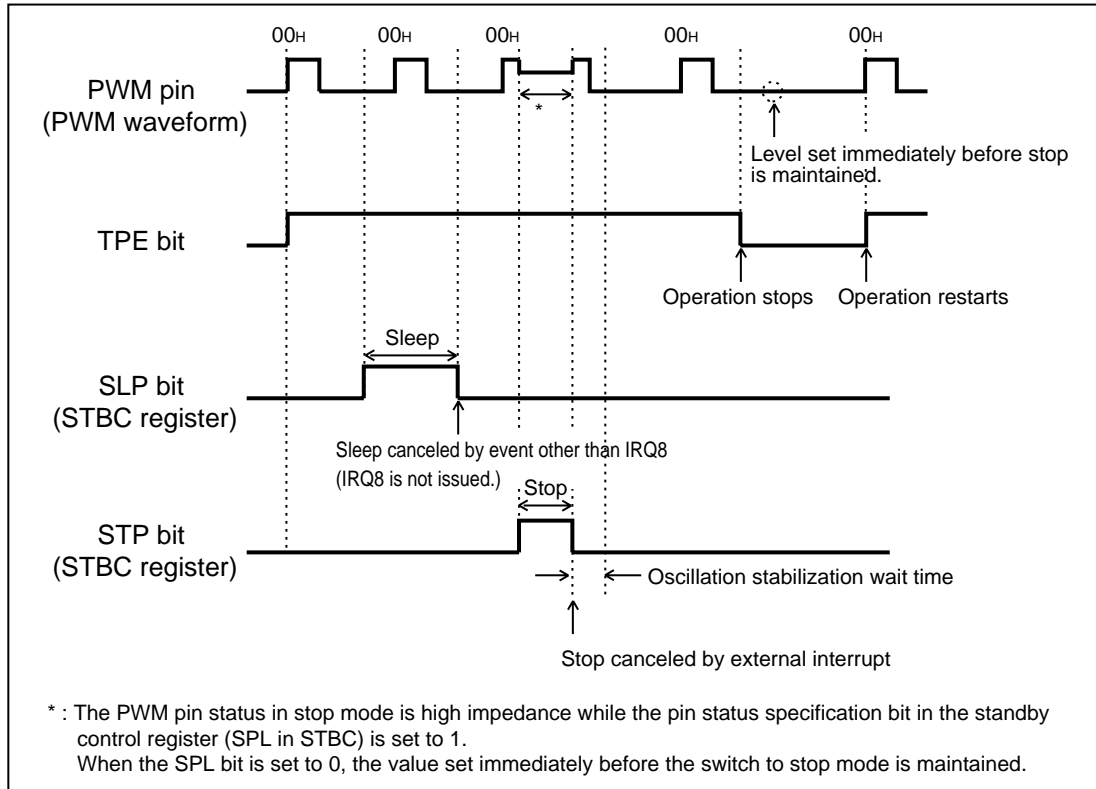
○ Interval timer operation

Figure 7.12-1 Counter Operation for Standby Mode and Intermediate Stop (Interval Function)



○ PWM timer operation

Figure 7.12-2 Counter Operation for Standby Mode and Intermediate Stop (PWM Timer Function)



7.13 Notes on Using the 2-Channel 8-Bit PWM Timer

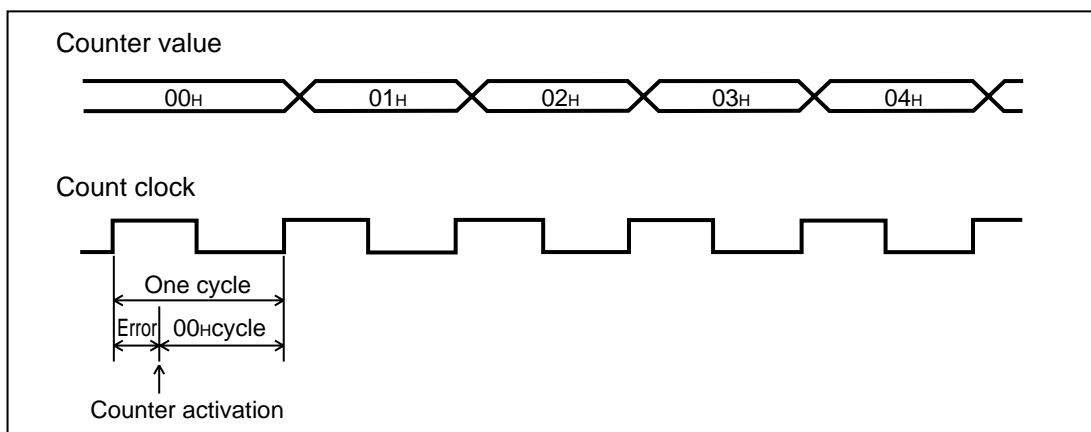
This section contains notes on using the 2-channel 8-bit PWM timer.

■ Notes on Using the 2-Channel 8-Bit PWM Timer

○ Error

Since counter activation by a program is not synchronized with the start of counting by the count clock, the 00H cycle may become shorter by one count clock cycle or less because of an error in the time required until a match between the counter value and COMR register value is detected. Figure 7.13-1 "Error in time before counting starts" shows this error.

Figure 7.13-1 Error in time before counting starts



○ Notes on setting the program

- Do not change the corresponding count clock cycle (SC11/SC10 or SC21/SC20 in CNTR1) while the PWM timer function or PWM timer function is operating (TPE1 = 1/TPE2 = 1 in CNTR2).
- Switch between the interval timer function and PWM timer function (PTX1/PTX2 in CNTR1) only while the counter is stopped (TPE1 = 1/TPE2 = 0 in CNTR2), interrupts are disabled (TIE1 = 0/TIE2 = 0 in CNTR2), and issued interrupt requests have been cleared (TIR1 = 0/TIR2 = 0 in CNTR2).
- Do not set CH12PWM mode (CH12 = 1 in CNTR3) and CH1 PWM timer operation (PTX1 = 1 in CNTR1) in CK12 mode (CK12 = 1 in CNTR2).
- Disable CH1 interrupt request output (TIE1 = 0 in CNTR2) when CH12PWM mode is set. Also, do not set CK12 mode.
- Return from interrupt handling is impossible while the interrupt request flag bit (TIR1/TIR2 in CNTR2) is set to 1 and interrupt request output is enabled (TIE1 = 1/TIE2 = 1 in CNTR2). Always clear the TIR bit after issuing an interrupt request.
- The TIR bit is not set if the counter value matches the COMR register value at the same time the counter has stopped (TPE1 = 0/TPE2 = 0 in CNTR2).

7.14 Program Examples of the 2-Channel 8-Bit PWM Timer (Interval Function)

This section contains program examples for the interval function for 8-bit timer mode and CK12 mode.

■ Program Example of 8-bit Timer Mode

○ Processing specifications

- The CH1 timer operates as an interval timer in 8-bit timer mode.
- Timer interrupts are generated repeatedly at 3.2 ms intervals.
- A rectangular wave inverted at specified intervals is output to the PWM pin.
- A COMR1 value to set an interval of about 3.2 ms for main clock oscillation of 12 MHz is shown below. An internal count clock of 64 t_{inst} (t_{inst} : Maximum clock speed (gear)) is assumed as the count clock.

$$\text{COMR1 register value} = 3.2 \text{ ms} / (64 \times 4 / 12 \text{ MHz}) - 1 = 149 \text{ (095}_{\text{H}})$$

○ Coding example (comply with Softune V1)

```

CNTR1 EQU 0027H ;Address of CNTR1
CNTR2 EQU 0028H ;Address of CNTR2
CNTR3 EQU 0029H ;Address of CNTR3
COMR1 EQU 002AH ;Address of COMR1
TPE1 EQU CNTR2:7 ;Definition of CH1 counter enable bit
TIR1 EQU CNTR2:3 ;Definition of CH1 interrupt request flag bit
ILR3 EQU 007E ;Address of interrupt level setting register
INT_V DSEG ABS ; [DATA SEGMENT]
      ORG 0FFEAH
IRQ8 DW WARI1 ;Interrupt vector setting
INT_V ENDS
;-----Main program-----
      CSEG ; [DATA SEGMENT]
      ;Stack pointer (SP) has been initialized.
      :
      CLRI ;Interrupt is disabled.
      CLRB TPE1 ;Counter operation is stopped.
      MOV ILR3,#11111101B ;Interrupt level setting (level 1)
      MOV COMR1,#095H ;Value to be compared to counter
                        ;value (interval)
      MOV CNTR1,#00001100B ;Interval time operation, 64 tinst selection
      MOV CNTR3,#00100000B ;PWM1 pin output enabled
      MOV CNTR2,#10000010B ;Counter start, interrupt request
                        ;output
      SETI ;Interrupt is enabled.
      :
;-----Interrupt program-----
WARI1 CLRB TIR1 ;Interrupt request flag is cleared.
      PUSHW A
      XCHW A,T ;A/T save
      PUSHW A
      :
      User processing
      :
      POPW A
      XCHW A,T ;A/T restoration
      POPW A
      RETI
      ENDS
;-----

```

■ Program Example of CK12 Mode

○ Processing specifications

- 3.2 ms is specified as the interval (rectangular output cycle: 6.4 ms) for the CH1 timer and no interrupts are used.
- Output from the CH1 timer is input to the CH2 timer as the count clock and the CH2 timer generates timer interrupts repeatedly at 64 ms intervals.
- A rectangular wave inverted at the CH2 interval is output to the PWM2 pin.
- A COMR1 value to set a CH1 interval of about 3.2 ms for main clock oscillation of 12 MHz is shown below. An internal count clock of $64 t_{\text{inst}}$ (t_{inst} : Maximum clock speed (gear)) is assumed as the count clock.

COMR1 register value = $3.2 \text{ ms} / (64 \times 4 / 12 \text{ MHz}) - 1 = 149 \text{ (095H)}$

A COMR2 register value to set a CH2 interval of about 64 ms with the rectangular output from the CH1 timer is shown below.

COMR2 register value = $64 \text{ ms} / (3.2 \times 2) \text{ ms} = 10 \text{ (00AH)}$

○ Coding example (comply with Softune V1)

```

CNTR1 EQU 0027H ;Address of CNTR1
CNTR2 EQU 0028H ;Address of CNTR2
CNTR3 EQU 0029H ;Address of CNTR3
COMR1 EQU 002AH ;Address of COMR1
COMR2 EQU 002BH ;Address of COMR2
TPE1 EQU CNTR2:7 ;Definition of CH1 counter enable bit
TPE2 EQU CNTR2:6 ;Definition of CH2 counter enable bit
TIR2 EQU CNTR2:2 ;Definition of CH2 interrupt request flag bit
ILR3 EQU 007E ;Address of interrupt level setting register
INT_V DSEG ABS ; [DATA SEGMENT]
      ORG 0FFEAH
IRQ9 DW WARI ;Interrupt vector setting
INT_V ENDS

;-----Main program-----
      CSEG ; [CODE SEGMENT]
      ;Stack pointer (SP) has been initialized.

      :
      CLRI ;Interrupt is disabled.
      CLRB TPE1 ;Counter operation is stopped.
      CLRB TPE2
      MOV ILR3,#11111101B ;Interrupt level setting (level 1)
      MOV COMR1,#095H ;Value to be compared to counter
                        value (interval)
      MOV COMR2,#00AH
      MOV CNTR1,#00001100B ;Interval time operation, 64 tinst selection
      MOV CNTR3,#01000000B ;PWM2 pin output enabled
      MOV CONT2,#11100001B ;Counter start, interrupt request output
      SETI ;Interrupt is enabled.
      :
;-----Interrupt program-----
WARI CLRB TIR2 ;Interrupt request flag is cleared.
      PUSHW A
      XCHW A,T ;A/T save
      PUSHW A
      :
      User processing
      :
      POPW A
      XCHW A,T ;A/T restoration
      POPW A
      RETI
      ENDS
;-----

```

7.15 Program Examples of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)

This section contains PWM timer function program examples for 8-bit PWM mode, 7-bit PWM mode, and CH12PWM mode.

■ Program Example of PWM Timer Function

○ Processing specifications

- The CH1 timer operates as a PWM timer in 8-bit PWM mode and a rectangular wave is output to the PWM1 pin.
- The CH2 timer operates as a PWM timer in 7-bit PWM mode (high-speed mode) and a rectangular wave is output to the PWM2 pin.
- A PWM wave with a duty ratio of 50% is generated, and then the duty ratio is changed to 25%.
- No interrupts are generated.
- When an internal count clock of $16 t_{\text{inst}}$ (t_{inst} : Maximum clock speed (gear)) is assumed as the count clock for the CH1 and CH2 timers with main clock oscillation of 12 MHz, the CH1 PWM wave cycle is 1.365 ms ($16 \times 2 / 12 \text{ MHz} \times 256$) and the CH2 PWM wave cycle is 6.82 ms ($16 \times 4 / 12 \text{ MHz} \times 128$).
- A COMR register value to set a duty ratio of 50% in 8-bit PWM mode is shown below:
COMR1 register value = $50/100 \times 256 = 128$ (80H)
- A COMR register value to set a duty ratio of 50% in 7-bit PWM mode is shown below:
COMR2 register value = $50/100 \times 128 = 64$ (40H)

○ Coding example (comply with Softune V1)

```

CNTR1 EQU 0027H ;Address of CNTR1
CNTR2 EQU 0028H ;Address of CNTR2
CNTR3 EQU 0029H ;Address of CNTR3
COMR1 EQU 002AH ;Address of COMR1
COMR2 EQU 002BH ;Address of COMR2
TPE1 EQU CNTR2:7 ;Definition of CH1 counter enable bit
TPE2 EQU CNTR2:6 ;Definition of CH2 counter enable bit
;-----Main program-----
CSEG ; [CODE SEGMENT]
:
CLRB TPE1 ;Counter operation is stopped.
CLRB TPE2
MOV COMR1,#80H ;Pulse H width specification, duty ratio 50%
MOV COMR2,#40H
MOV CNTR1,#11011010B ;PWM timer operation, 8/7-bit PWM mode,
; 16 tinst selection
MOV CNTR3,#01000000B ;PWM1/PWM2 pin output enabled
MOV CNTR2,#11000000B ;Counter start, interrupt request output
; disabled
:
:
MOV COMR1,#40H ;Duty ratio is changed to 25% (becomes valid
; in the next PWM wave cycle)
MOV COMR2,#20H
:
ENDS
;-----
END

```

■ Program Example of CH12PWM Mode

○ Processing specifications

- A PWM wave with duty ratio of 50% is generated and then the duty ratio is changed to 25%.
- When an internal count clock of 16 t_{inst} (t_{inst} : Maximum clock speed (gear)) is assumed as the count clock for the CH1 and CH2 timers with main clock oscillation of 12 MHz and 128 (80_H) is set in the COMR2 register, the PWM wave cycle is 682.6 ms ($16 \times 4/12 \text{ MHz} \times 128$).
- A COMR1 register value to set a duty ratio of 50% is shown below:
- COMR1 register value = $50/100 \times 128 = 64$ (040_H)

○ Coding example (comply with Softune V1)

```

CNTR1 EQU 0027H ;Address of CNTR1
CNTR2 EQU 0028H ;Address of CNTR2
CNTR3 EQU 0029H ;Address of CNTR3
COMR1 EQU 002AH ;Address of COMR1
COMR2 EQU 002BH ;Address of COMR2

TPE1 EQU CNTR2:7 ;Definition of CH1 counter enable bit
TPE2 EQU CNTR2:6 ;Definition of CH2 counter enable bit
;-----Main program-----
CSEG ; [CODE SEGMENT]
:
CLRB TPE1 ;Counter operation is stopped.
CLRB TPE2
MOV COMR1,#40H ;Pulse L width specification, duty ratio 50%
MOV COMR2,#80H ;Pulse one cycle specification
MOV CNTR1,#00001010B ;16 tinst selection
MOV CNTR3,#01010000B ;PWM2 pin output enabled
MOV CNTR2,#11000000B ;Counter starts, interrupt request output
                        disabled
:
:
MOV COMR1,#20H ;Duty ratio is changed to 25%
                        (takes effect in the next PWM wave cycle)
:
ENDS
;-----

```


CHAPTER 8 EXTERNAL INTERRUPT CIRCUIT (LEVEL)

This chapter explains the functions and operation of the external interrupt circuit (level).

- 8.1 "Overview of the External Interrupt Circuit (Level)"
- 8.2 "Configuration of the External Interrupt Circuit"
- 8.3 "Pins of the External Interrupt Circuit"
- 8.4 "Registers of the External Interrupt Circuit"
- 8.5 "Interrupt of the External Interrupt Circuit"
- 8.6 "Operations of the External Interrupt Circuit"
- 8.7 "Program Example of the External Interrupt Circuit"

8.1 Overview of the External Interrupt Circuit (Level)

The external interrupt circuit detects the level of signal input to eight external interrupt pins, and issues a single interrupt request to the CPU.

■ Function of External Interrupt Circuit (Level Detection)

The function of the external interrupt circuit is to detect L level signal input to the external interrupt pin, and to issue an interrupt request to the CPU. This interrupt enables recovery from standby mode and transfer to normal operation (main RUN state or sub-RUN state).

- External interrupt pin: 8 (P30/ $\overline{\text{INT0}}$ to P37/ $\overline{\text{INT7}}$)
- Cause of external interrupt: L level signal input to external interrupt pin
- Interrupt control: Enable and disable external interrupt input from external interrupt control register (EIE).
- Interrupt flag: Detects L level from external interrupt request flag bit of external interrupt flag register (EIF).
- Interrupt request: Issued by ORing of individual external interrupt causes (IRQ0).

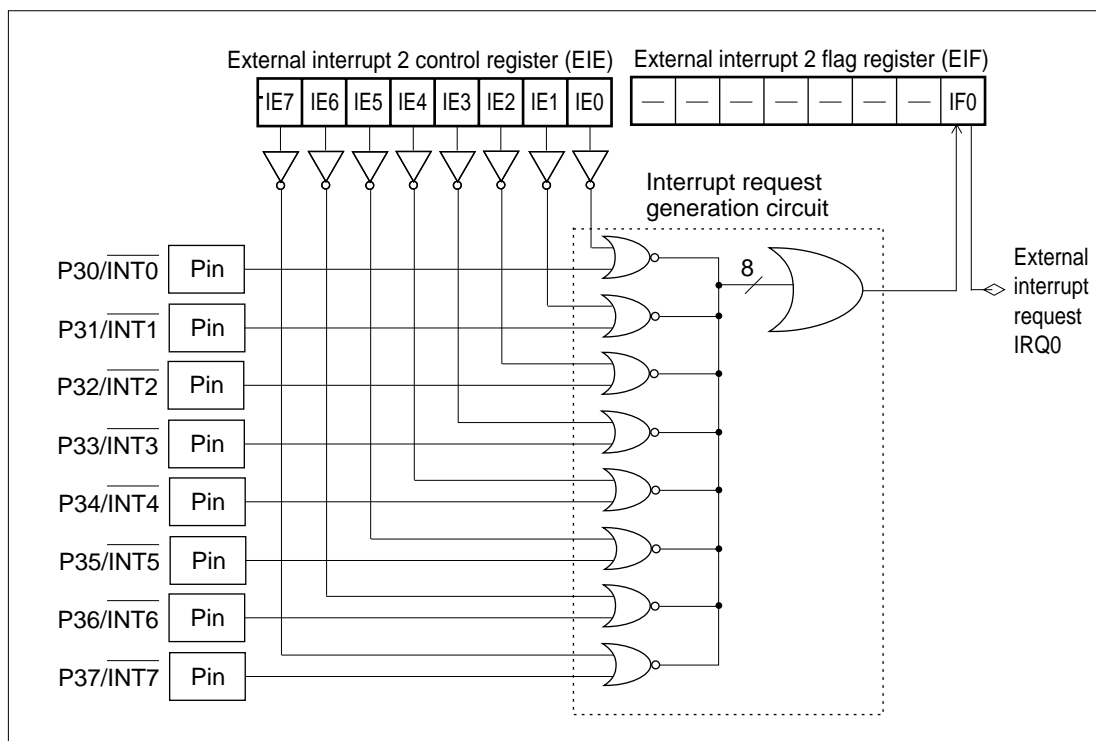
8.2 Configuration of the External Interrupt Circuit

The external interrupt circuit contains the following three blocks:

- Interrupt request generation circuit
- External interrupt control register (EIE)
- External interrupt flag register (EIF)

■ Block Diagram of External Interrupt Circuit

Figure 8.2-1 Block Diagram of External Interrupt Circuit



○ Interrupt request generation circuit

The interrupt request generation circuit issues an interrupt request signal based on the signal input to the external interrupt pins ($\overline{\text{INT0}}$ to $\overline{\text{INT7}}$) and the external interrupt input enable bit.

○ External interrupt control register (EIE)

The external interrupt input enable bits (IE0 to IE7) enable or disable L level input from the corresponding external interrupt pins.

○ External interrupt flag register (EIF)

The external interrupt flag register saves and clears the interrupt request signal issued based on the external interrupt request flag bit (IF0).

CHAPTER 8 EXTERNAL INTERRUPT CIRCUIT (LEVEL)

○ Interrupt cause of external interrupt circuit

An interrupt request is issued when an L level signal is input to an external interrupt pin (either IRQ0 or $\overline{\text{INT0}}$ to $\overline{\text{INT7}}$), and when the external interrupt input enable bit corresponding to the pin is 1.

8.3 Pins of the External Interrupt Circuit

This section describes the pins related to the external interrupt circuit and contains a pin block diagram.

■ Pins Related to External Interrupt Circuit

There are eight external interrupt pins that are related to the external interrupt circuit.

○ P30/ $\overline{\text{INT0}}$ to P37/ $\overline{\text{INT7}}$

These external interrupt pins (P30/ $\overline{\text{INT0}}$ to P37/ $\overline{\text{INT7}}$) function as external interrupt input (hysteresis input) and also as general-purpose input and output ports.

Pins P30/ $\overline{\text{INT0}}$ to P37/ $\overline{\text{INT7}}$ set the pins corresponding to the port direction register (DDR3) as the input port, and function as the external interrupt pins ($\overline{\text{INT0}}$ to $\overline{\text{INT7}}$) after external interrupt input is enabled from the external interrupt control register (EIE). The state of the pins can be read from the port data register (PDR3) when it set to the input port.

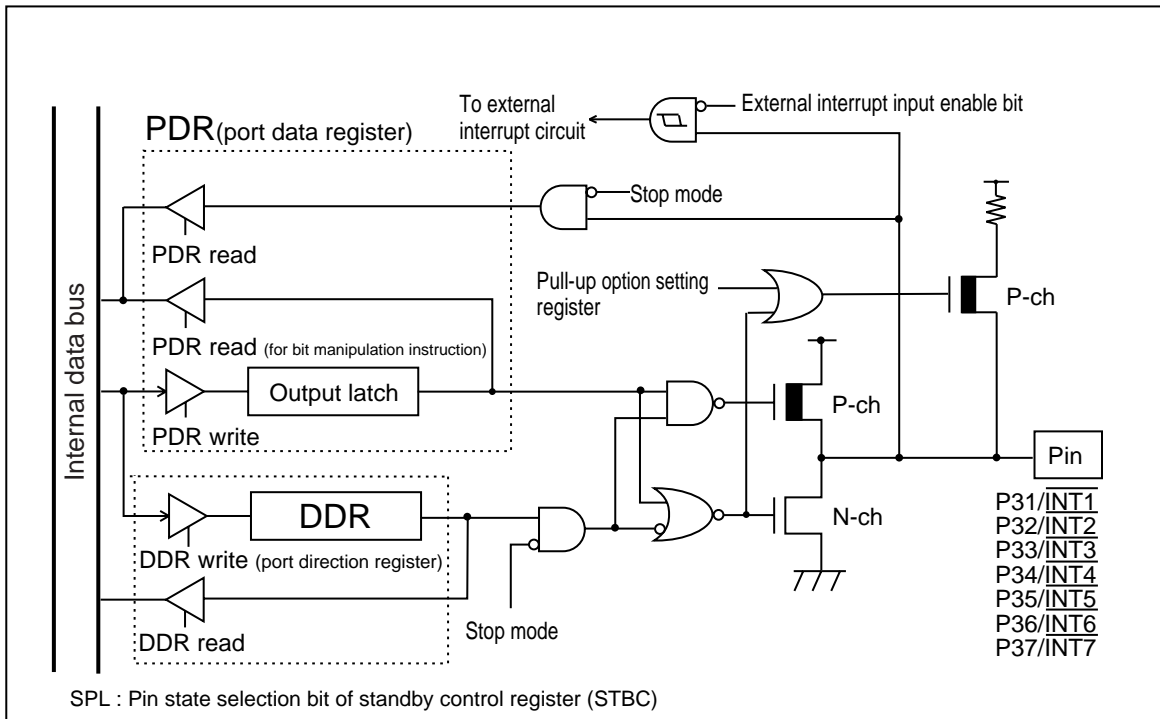
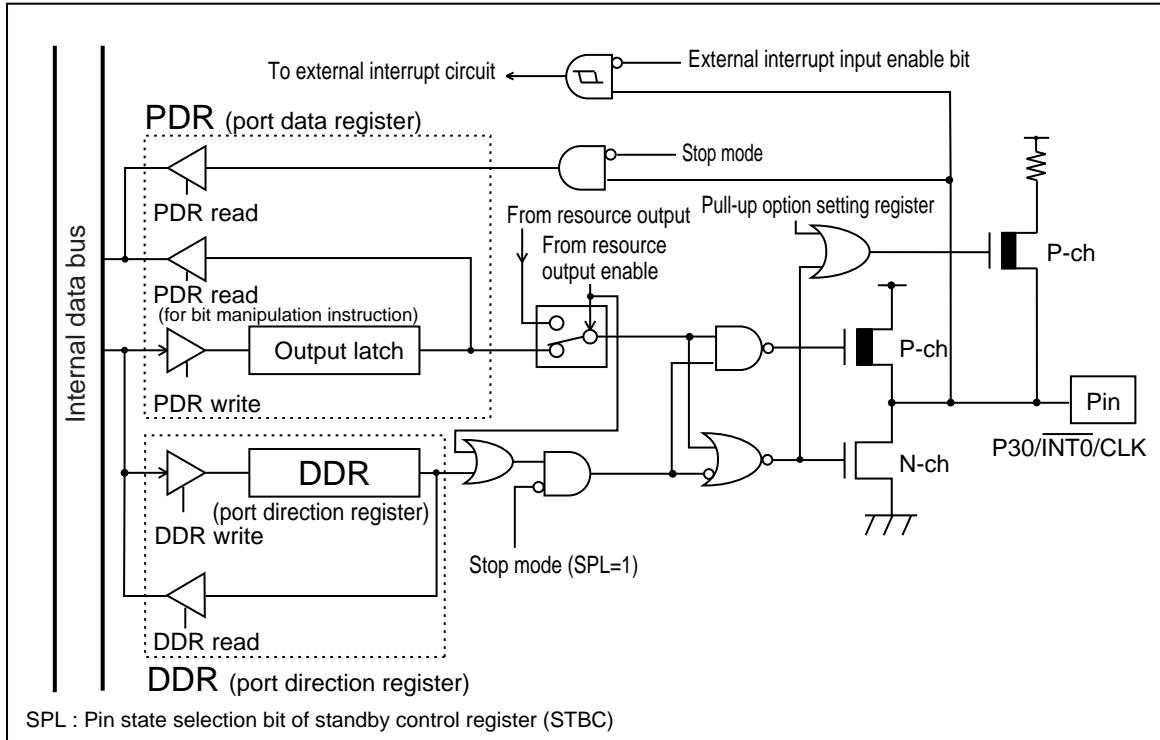
Table 8.3-1 "Pins Related to the External Interrupt Circuit" shows the pins related to the external interrupt circuit.

Table 8.3-1 Pins Related to the External Interrupt Circuit

External interrupt pin	Used as external interrupt input (interrupt input enabled)	Used as general-purpose input and output port (interrupt input disabled)
P30/ $\overline{\text{INT0}}$	$\overline{\text{INT0}}$ (EIE:IE0=1, DDR3:bit0=0)	P30 (EIE:IE0=0)
P31/ $\overline{\text{INT1}}$	$\overline{\text{INT1}}$ (EIE:IE1=1, DDR3:bit1=0)	P31 (EIE:IE1=0)
P32/ $\overline{\text{INT2}}$	$\overline{\text{INT2}}$ (EIE:IE2=1, DDR3:bit2=0)	P32 (EIE:IE2=0)
P33/ $\overline{\text{INT3}}$	$\overline{\text{INT3}}$ (EIE:IE3=1, DDR3:bit3=0)	P33 (EIE:IE3=0)
P34/ $\overline{\text{INT4}}$	$\overline{\text{INT4}}$ (EIE:IE4=1, DDR3:bit4=0)	P34 (EIE:IE4=0)
P35/ $\overline{\text{INT5}}$	$\overline{\text{INT5}}$ (EIE:IE5=1, DDR3:bit5=0)	P35 (EIE:IE5=0)
P36/ $\overline{\text{INT6}}$	$\overline{\text{INT6}}$ (EIE:IE6=1, DDR3:bit6=0)	P36 (EIE:IE6=0)
P37/ $\overline{\text{INT7}}$	$\overline{\text{INT7}}$ (EIE:IE7=1, DDR3:bit7=0)	P37 (EIE:IE7=0)

■ Block Diagram of the Pins Related to the External Interrupt Circuit

Figure 8.3-1 Block Diagram of the Pins Related to the External Interrupt Circuit



Reference:

If pull-up resistor is selected in the pull-up option register, the state of pins in stop mode (SPL = 1) becomes H level (pull-up state) instead of high impedance. However, during reset, the pull-up state becomes invalid, and the state of the pins becomes Hi-Z.

■ Relationship between Interrupt Enable Bit and External Interrupt Pin of External Interrupt Circuit

Table 8.3-2 "Relationship between Interrupt Enable Bit and External Interrupt Pin" shows the relationship between the interrupt enable bit and the external interrupt pin.

Table 8.3-2 Relationship between Interrupt Enable Bit and External Interrupt Pin

Register	Bit name		External interrupt pin
EIE	bit0	IE0	$\overline{\text{INT0}}$
	bit1	IE1	$\overline{\text{INT1}}$
	bit2	IE2	$\overline{\text{INT2}}$
	bit3	IE3	$\overline{\text{INT3}}$
	bit4	IE4	$\overline{\text{INT4}}$
	bit5	IE5	$\overline{\text{INT5}}$
	bit6	IE6	$\overline{\text{INT6}}$
	bit7	IE7	$\overline{\text{INT7}}$

8.4 Registers of the External Interrupt Circuit

This section explains the register related to the external interrupt circuit.

■ Register Related to External Interrupt Circuit

Figure 8.4-1 Register Related to External Interrupt Circuit

EIE (External interrupt control register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 C _H	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
EIF (External interrupt flag register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 D _H	—	—	—	—	—	—	—	IF0	XXXXXXX0 _B
								R/W	
R/W :Readable and writable									
— :Unused									
X :Undefined									

8.4.1 External Interrupt Control Register (EIE)

The external interrupt control register (EIE) enables and disables the interrupt input on external interrupt pins $\overline{\text{INT0}}$ to $\overline{\text{INT7}}$.

■ External Interrupt Control Register (EIE)

Figure 8.4-2 External Interrupt Control Register (EIE)

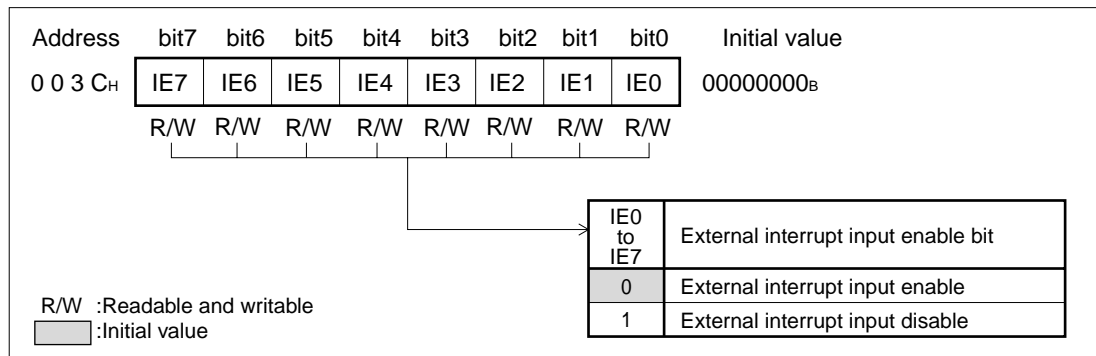


Table 8.4-1 Relationship between Bits and External Interrupt Pin of External Interrupt Control Register (EIE)

	Bit name	External interrupt pin
bit7	IE7	$\overline{\text{INT7}}$
bit6	IE6	$\overline{\text{INT6}}$
bit5	IE5	$\overline{\text{INT5}}$
bit4	IE4	$\overline{\text{INT4}}$
bit3	IE3	$\overline{\text{INT3}}$
bit2	IE2	$\overline{\text{INT2}}$
bit1	IE1	$\overline{\text{INT1}}$
bit0	IE0	$\overline{\text{INT0}}$

CHAPTER 8 EXTERNAL INTERRUPT CIRCUIT (LEVEL)

Table 8.4-2 Functions of the External Interrupt Control Register (EICR) Bits

Bit name		Function
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0	IE0 to IE7: External interrupt input enable bit	<ul style="list-style-type: none">• These bits enable and disable the interrupt input of external interrupt pins $\overline{INT0}$ to $\overline{INT7}$.• When these bits are set to 1, the corresponding external interrupt pins function as external interrupt input pins, and receive external interrupt input.• When these bits are set to 0, the corresponding external interrupt pins function as a general-purpose port, and do not receive external interrupt input. References: <ul style="list-style-type: none">• When using an external interrupt pin, write 0 to the corresponding bit of the port direction register (DDR3), and set the pin to input.• The external interrupt pin state can be read directly from the port data register (PDR3), regardless of the external interrupt input enable state.

8.4.2 External Interrupt Flag Register (EIF)

The external interrupt flag register (EIF) detects level interrupts and clears interrupt requests.

■ External Interrupt Flag Register (EIF)

Figure 8.4-3 External Interrupt Flag Register (EIF)

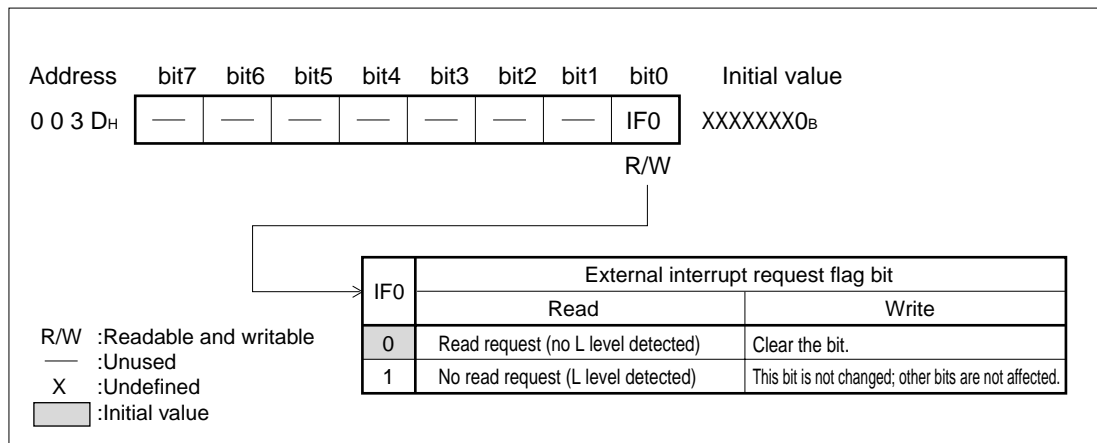


Table 8.4-3 Functions of Bit of External Interrupt Flag Register (EIF) Bits

Bit name		Function
bit7 bit6 bit5 bit4 bit3 bit2 bit1	Unused bit	<ul style="list-style-type: none"> Read values are undefined. Writing has no effect on operation.
bit0	IF0: External interrupt request flag bit	<ul style="list-style-type: none"> When an L level signal is input to the external interrupt pins ($\overline{\text{INT0}}$ to $\overline{\text{INT7}}$) with external interrupt input enabled, the external interrupt request flag bit is set to 1. Writing 0 to this bit clears it. Writing 1 to this bit does not change it, and has no effect on other bits. <p>Note: The external interrupt input enable bits (EIE: IE0 to IE7) of the external interrupt control register only disable external interrupt input. Interrupt requests continue until the IF0 bit is cleared to 0.</p>

8.5 Interrupt of the External Interrupt Circuit

An L level input signal input to the external interrupt pins causes an external interrupt circuit interrupt.

■ Interrupts Issued during External Interrupt Circuit Operation

When an L level signal is input to the external interrupt pins with interrupt input enabled, the external interrupt request flag bit (EIF: IF0) is set to 1, and an interrupt request (IRQ0) is issued to the CPU. Write 0 to the IF0 bit in the interrupt processing routine, and clear the interrupt request.

When the external interrupt request flag bit (IF0) is set to 1 even though external interrupt input is disabled, interrupt requests will be issued continuously in response to the settings of the interrupt enable bits (IE0 to IE7) of the external interrupt control register (EIE) until the IF0 bit is cleared to 0. The IF0 bit must therefore be cleared.

Also, if an external interrupt pin is at the L level, the IF0 bit is set immediately even though it is cleared when external interrupt input is enabled. Consequently, disable external interrupt input or release the external interrupt cause if necessary.

After releasing a reset, clear the IF0 bit before enabling CPU interrupts.

Note:

- Since an L level input to any of the external interrupt pins ($\overline{\text{INT0}}$ to $\overline{\text{INT7}}$) issues the same interrupt request (IRQ0), it is necessary to determine to which pin the external interrupt input applies. Do this by reading the port data register (PDR3) before changing the input to the H level.
- A stop mode interrupt can be released only by the external interrupt circuit.

■ Register and Vector Table Related to Interrupt of External Interrupt Circuit

Table 8.5-1 Register and Vector Table Related to the Interrupt of External Interrupt Circuit

Interrupt name	Interrupt level setting register			Address of vector table	
	Register	Setting bit		Higher	Lower
IRQ0	ILR1(007C _H)	L01(bit1)	L00(bit0)	FFFA _H	FFFB _H

See Section 3.4.2 "Processing During Interrupt Operation" for details of the interrupt operation.

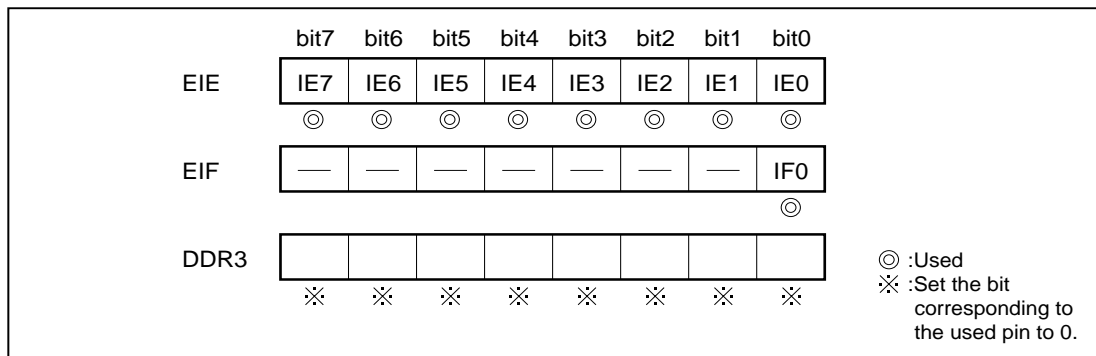
8.6 Operations of the External Interrupt Circuit

The external interrupt circuit detects that an external interrupt pin is L level, and issues an interrupt request to the CPU.

■ Operations of the External Interrupt Circuit

The settings in Figure 8.6-1 "Setting of External Interrupt Circuit" are required in an instruction that uses the external interrupt circuit.

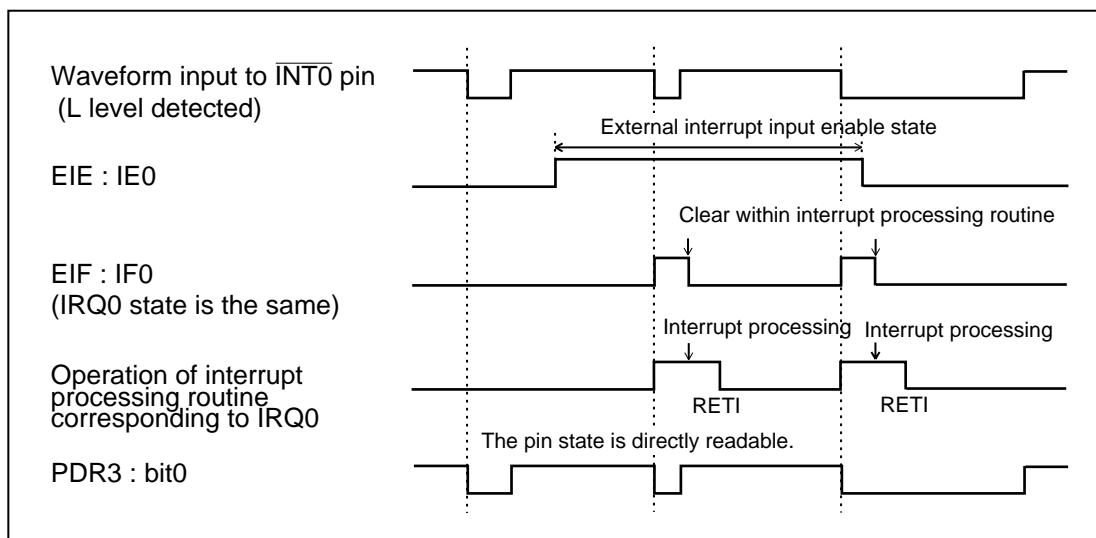
Figure 8.6-1 Setting of External Interrupt Circuit



When an L level signal is input to the external interrupt pins corresponding to $\overline{\text{INT0}}$ to $\overline{\text{INT7}}$ when external interrupt input (one of IE0 to IE7) is enabled, an IRQ0 interrupt request is issued to the CPU.

Figure 8.6-2 "Operation of External Interrupts (INT0)" shows the operation of the external interrupt circuit (when INT0 pin used).

Figure 8.6-2 Operation of External Interrupts (INT0)



Reference:

The pin state can be read directly from the port data register (PDR3) even though an external interrupt pin is used for external interrupt input.

8.7 Program Example of the External Interrupt Circuit

This section presents examples of external interrupt circuit programs.

■ Program Example of the External Interrupt Circuit

○ Processing specification

Detects the L level when it is input to the $\overline{\text{INT0}}$ pin, and issues an interrupt.

○ Coding example (comply with Softune V1)

```

DDR3    EQU    000DH           ;Address of port direction register
EIE      EQU    003CH           ;Address of external interrupt
                                ;control register
EIF      EQU    003DH           ;Address of external interrupt flag
                                ;register
IF0      EQU    EIF0           ;Definition of external interrupt
                                ;request flag bit
ILR1     EQU    007CH           ;Address of interrupt level setting
                                ;register
INT_V    DSEG    ABS           ;[DATA SEGMENT]
          ORG     0FFF2H
IRQ0     DW      WARI           ;Interrupt vector setting
INT_V    ENDS
;-----Main program-----
CSEG                                           ;[CODE SEGMENT]
          ;Initialization of the stack pointer
          ;(SP) is assumed.
          :
          CLRI                                           ;Interrupt disable
          CLRB     IF0                                     ;Clear external interrupt request
                                ;flag.
          MOV      ILR1,#11111110B ;Set interrupt level to 2.
          MOV      DDR3,#00000000B ;Set INT0 pin to input.
          MOV      EIE,#00000001B ;Enable external interrupt input of
                                ;INT0 pin.
          SETI                                           ;Interrupt enable
          :
;-----Interrupt processing routine-----
WARI     MOV      EIE,#00000000B ;Disable external interrupt input of
                                ;INT0 pin.
          CLRB     IF0                                     ;Clear external interrupt request
                                ;flag.
          PUSHW    A
          XCHW     A,T
          PUSHW    A
          :
          User processing
          :
          POPWA    A
          XCHW     A,T
          POPWA    A
          RETI
          ENDS
;-----
          END

```


CHAPTER 9 PARALLEL PORTS

This chapter explains the functions and operation of the parallel ports.

- 9.1 "Overview of the Parallel Ports"
- 9.2 "Configuration of the Parallel Port"
- 9.3 "Pins of the Parallel Port"
- 9.4 "Registers of the Parallel Port"
- 9.5 "Interrupts of the Parallel Port"
- 9.6 "Operation of the Parallel Port"

9.1 Overview of the Parallel Ports

A parallel port enables data to be exchanged with an external host CPU or peripheral device asynchronously via an external 8-bit system data bus.

■ Parallel Port Functions

A parallel port has functions for exchanging parallel data (8 bits) with another CPU or peripheral device.

- An 8-bit data bus can be used.
- A buffer function is available for independent input and output.
- Asynchronous data access from an external device is possible.
- The I/O buffer status can be output externally.

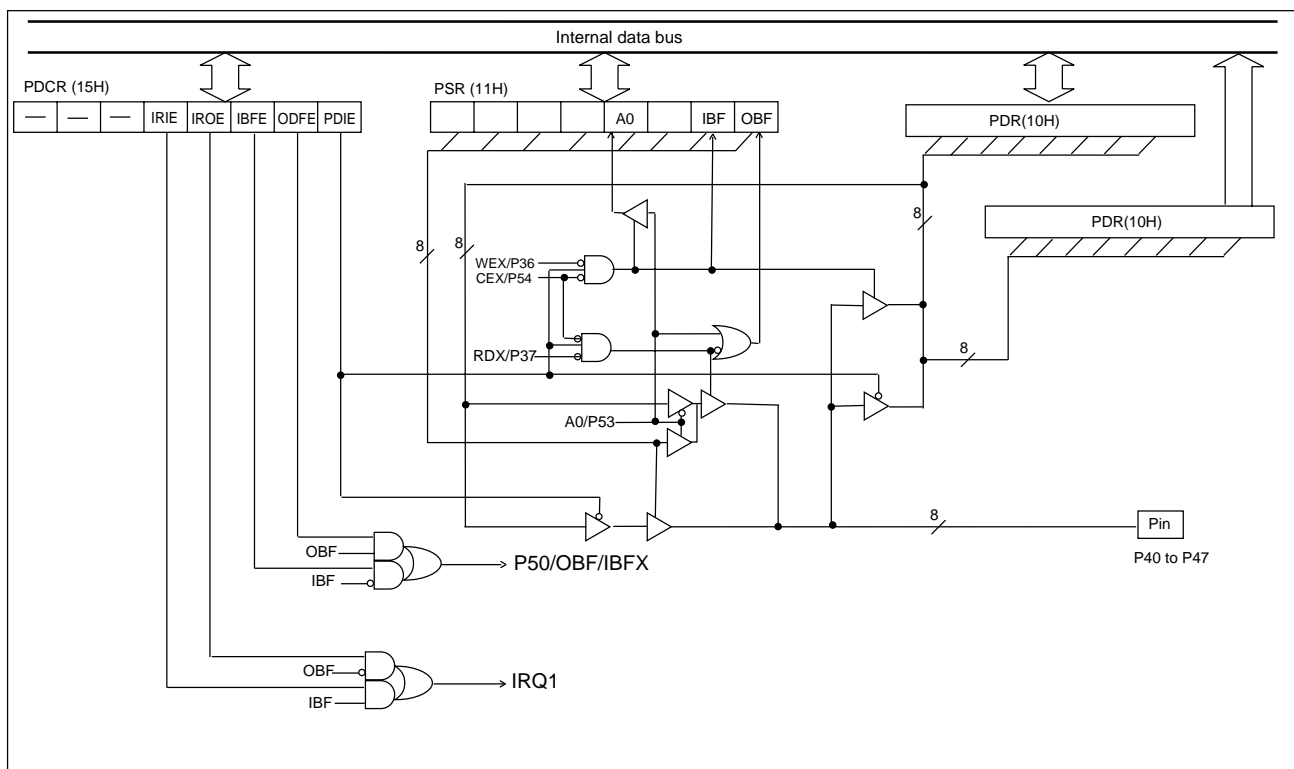
9.2 Configuration of the Parallel Port

A parallel port consists of the following three blocks:

- Parallel port data control register (PDCR)
- Parallel port status register (PSR)
- Parallel port data register (PDR)

■ Block Diagram of Parallel Port

Figure 9.2-1 Block Diagram of Parallel Port



○ Parallel port data control register (PDCR)

This register enables or disables parallel port interrupts and switches the function of the port between parallel port and general-purpose port.

○ Parallel port status register (PSR)

This register indicates the I/O buffer status as status data. The status can be output when data is output.

○ Parallel port data register (PDR)

This register is used as a data I/O buffer register for parallel port operation.

9.3 Pins of the Parallel Port

This section explains the parallel port pins and provides a block diagram of them.

■ parallel port pins

The parallel port pins are the data I/O pins (P40/D0 to P47/D7), external data write pin (P36/WEX), external data read pin (P37/RDX), parallel port device select pin (P54/CEX), data select pin (P53/A0) and status flag output terminals (P50/OBF, IBFX). Switching between these pins is accomplished by the setting of the parallel port operation enable bit (PDIE bit of PDCR).

P40/D0 to P47/D7:

These pins serve both as general-purpose I/O ports (P40 to P47) and as parallel port data I/O pins (D0 to D7). When parallel port operation is enabled (PDIE bit of PDCR is 1), these pins function as parallel port data I/O pins (D0 to D7) regardless of the value of the corresponding port direction register.

P36/WEX:

This pin serves as both a general-purpose I/O port (P36) and an external parallel port data write pin (WEX). When parallel port operation is enabled (PDIE bit of PDCR is 1), this pin functions as an external parallel port data write pin (WEX). If parallel port operation is enabled, set the corresponding port direction register to the input mode (set bit 6 of DDR3 to 0).

P37/RDX:

This pin serves as both a general-purpose I/O port (P37) and an external parallel port data read pin (RDX). When parallel port operation is enabled (PDIE bit of PDCR is 1), this pin functions as an external parallel port data read pin (RDX). If parallel port operation is enabled, set the corresponding port direction register to the input mode (set bit 7 of DDR3 to 0).

P54/CEX:

This pin serves as both a general-purpose I/O port (P54) and a parallel port device select pin (CEX). When parallel port operation is enabled (PDIE bit of PDCR is 1), this pin functions as a parallel port device select pin (CEX) regardless of the value of the corresponding port data register.

P53/A0:

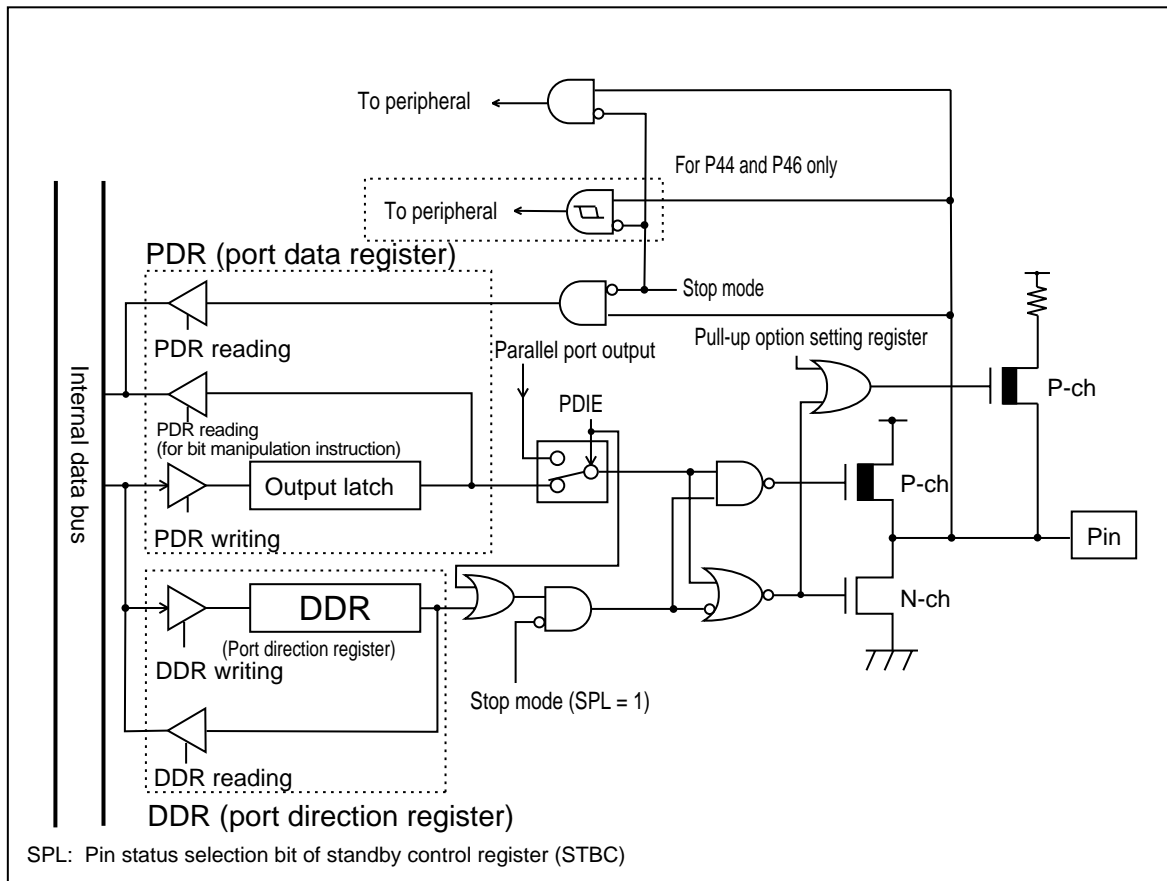
This pin serves as both a general-purpose I/O port (P53) and a parallel port data select pin (A0). When parallel port operation is enabled (PDIE bit of PDCR is 1), this pin functions as a parallel port data select pin (A0) regardless of the value of the corresponding port data register.

P50/OBF and IBFX:

This pin serves as both a general-purpose I/O port (P50) and parallel port status flag output pins (OBF and IBFX). When parallel port operation is enabled (PDIE bit of PDCR is 1), this pin functions as parallel port status flag output pins (OBF and IBFX) regardless of the value of the corresponding port direction register.

■ Block Diagram of Parallel Port Pin

Figure 9.3-1 Block Diagram of Parallel Port Pin



○ Pull-up

When the pull-up option setting register is set to enable use of the pull-up resistor, the pin status in stop mode (SPL = 1) is not high impedance status, but H level (pull-up status). However, the pull-up resistor has no effect in the reset state, and the pin status is high impedance.

9.4 Registers of the Parallel Port

This section explains the parallel port registers.

■ Parallel Port Registers

Figure 9.4-1 Parallel Port Registers

Parallel port data register (PDR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDR	0 0 1 0 _H	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0	XXXXXXXX _H
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Parallel port status register (PSR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PSR	0 0 1 1 _H	UF4	UF3	UF2	UF1	A0	UF0	IBF	OBF	XXXX0XXX _B
		R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	

Parallel port data control register (PDCR)

	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDCR	0 0 1 5 _H	-	-	-	IRIE	IROE	IBFE	OBFE	PDIE	XXX00000 _B
					R/W	R/W	R/W	R/W	R/W	

R/W : Readable and writable

R : Read only

- : Unused

X : Undefined

9.4.1 Parallel Port Data Register (PDR)

The parallel port data register (PDR) operates as a parallel port data I/O buffer.

■ Parallel Port Data Register (PDR)

Figure 9.4-2 Parallel Port Data Register (PDR)

Parallel port data register (PDR)										
	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
PDR	0 0 1 0 _H	PDB7	PDB6	PDB5	PDB4	PDB3	PDB2	PDB1	PDB0	XXXXXXXX _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Readable and writable

○ Output data buffer

During parallel port operation (PDIE bit of PDCR is 1), the PDR retains the data to be output to an external system bus. The data retained in the PDR is output to pins P40/D0 to P47/D7 when ORing of P36/WEX and P54/CEX results in 0.

○ Input data buffer

During parallel port operation (PDIE bit of PDCR is 1), the PDR retains the data input from pins P40/D0 to P47/D7. The values at the pins are latched when ORing of P36/WEX and P54/CEX results in 0.

The parallel port data register (PDR) uses the same address for the input and output buffers. When this register is accessed for reading during parallel port operation, input data is read. When this register is accessed for writing during parallel port operation, output data is written.

9.4.2 Parallel Port Status Register (PSR)

The parallel port status register (PSR) indicates the status of the data bus.

■ Parallel Port Status Register (PSR)

Figure 9.4-3 Parallel Port Status Register (PSR)

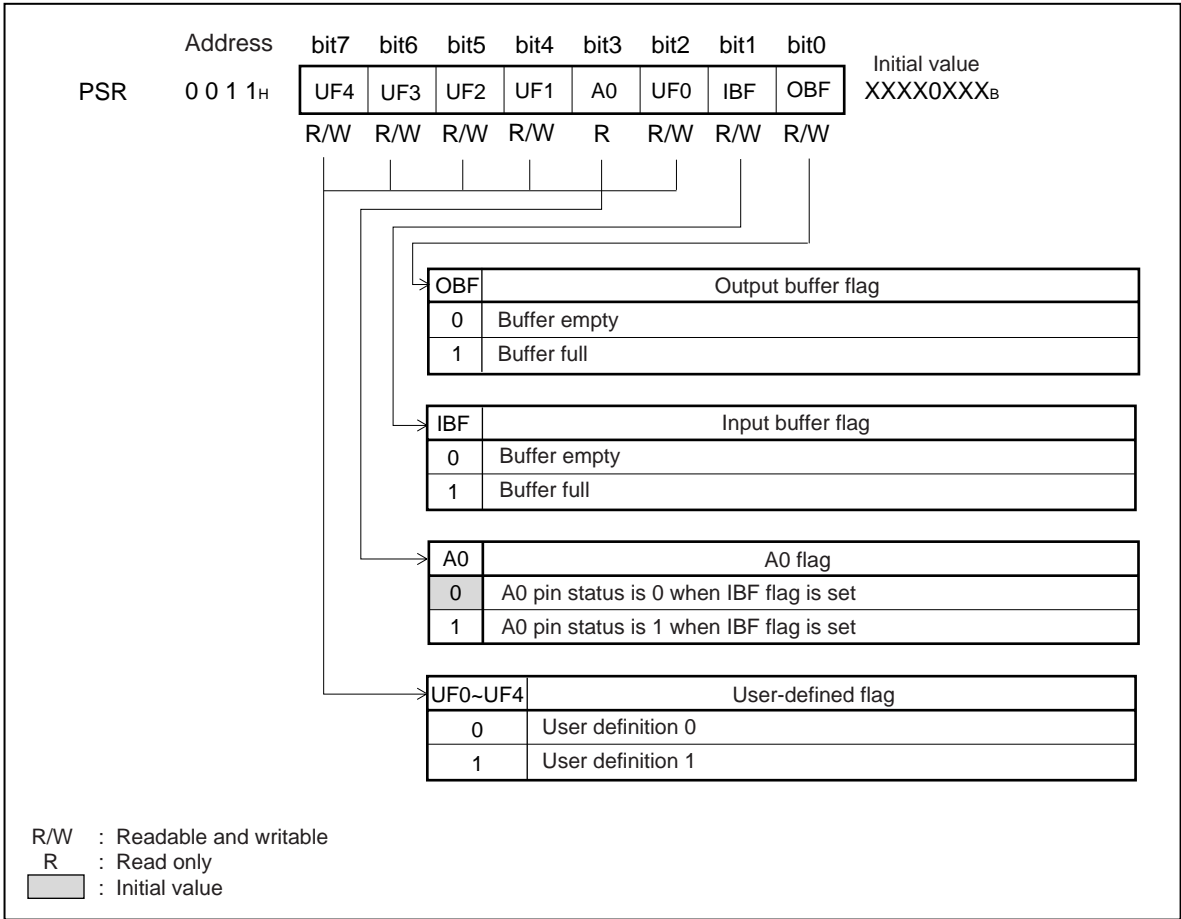


Table 9.4-1 Bit Functions of the Parallel Port Status Register (PSR)

Bit name		Function
bit7 to 4	UF4 to UF1: User-defined flag bits	<ul style="list-style-type: none"> These bits can be defined as flags by the user. Specifying whether they are set and cleared is optional by the program.
bit3	A0: A0 flag bit	<ul style="list-style-type: none"> This bit indicates the status of A0/P53 set when the IBF flag is set (WEX/P36 = 0, CEX/P54 = 0, and PDIE = 1).
bit2	UF0: User-defined flag bit	<ul style="list-style-type: none"> These bits can be defined as flags by the user. Specifying whether they are set and cleared is optional by the program.
bit1	IBF: Input buffer flag bit	<ul style="list-style-type: none"> This bit is set to 1 when external data is written to the input buffer (PDR). When this bit and the IRIE bit of PDCR are 1, an interrupt request is output to the CPU. When this bit and the IBIE bit of PDCR are 1, 0 is output to the external pin (P50/IBFX). In a write operation, this bit is cleared when 0 is written, but is not affected when 1 is written.
bit0	OBF: Output buffer flag bit	<ul style="list-style-type: none"> This bit is cleared to 1 when data is externally read from the output buffer (PDR). When this bit is cleared and the IROE bit of PDCR is 1, an interrupt request is output to the CPU. When this bit and the OBFE bit of PDCR are 1, 0 is output to the external pin (P50/OBF). In a write operation, this bit is set when 1 is written, but is not affected when 0 is written.

Note:

The PSR parallel port register also functions as direction register DDR4 in the general-purpose port. When the parallel port data control register (PDIE bit of PDCR) is used to switch between the general-purpose port and parallel port functions, the registers hold the latest values. (Only bit A0 is initialized when the parallel port function is set to on.)

9.4.3 Parallel Port Data Control Register (PDCR)

The parallel port data control register (PDCR) switches the function of ports between general-purpose ports and parallel ports, and enables or disables interrupts to the CPU and external pins.

■ Parallel Port Data Control Register (PDCR)

Figure 9.4-4 Parallel Port Data Control Register (PDCR)

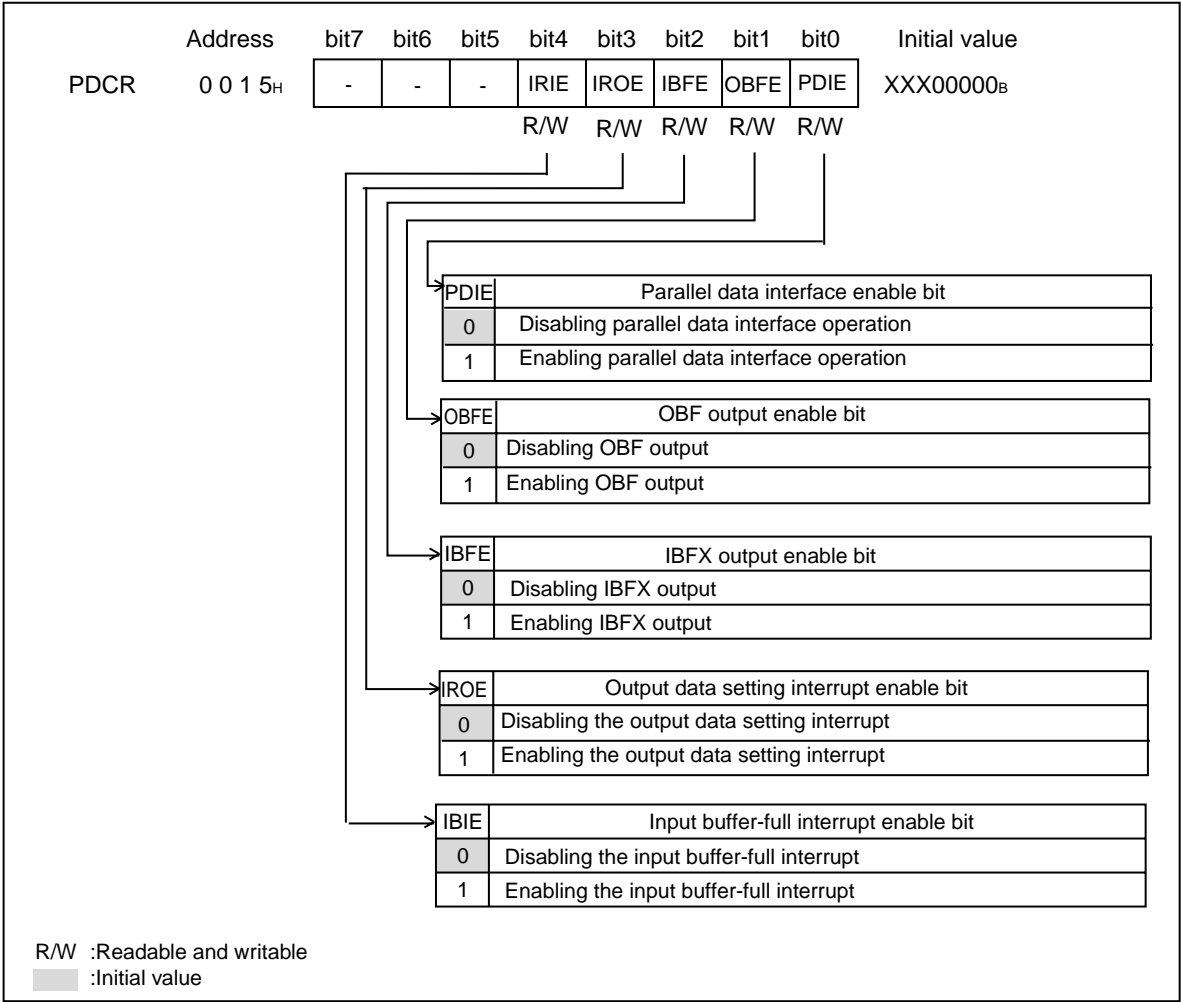


Table 9.4-2 Bit Functions of the Parallel Port Data Control Register (PDCR)

Bit name		Function
bit7 to 5	Unused bits	<ul style="list-style-type: none"> In a read, the status of these bits is undefined. In a write, these bits do not affect operation.
bit4	IRE: Input buffer-full interrupt enable bit	<ul style="list-style-type: none"> If this bit is 1 when the IBF flag is set, an interrupt request is output to the CPU.
bit3	IROE: Output data setting interrupt enable bit	<ul style="list-style-type: none"> If this bit is 1 when the OBF flag is cleared, an interrupt request is output to the CPU.
bit2	IBFE: IBFX output enable bit	<ul style="list-style-type: none"> If this bit is 1 when the IBF flag is cleared, 1 is output to the IBFX pin.
bit1	OBFE: OBF output enable bit	<ul style="list-style-type: none"> If this bit is 1 when the OBF flag is set, 1 is output to the OBF pin.
bit0	PDIE: Parallel data interface enable bit	<ul style="list-style-type: none"> When this bit is 1, related ports (P36, P37, P40 to P47, P50, P53, and P54) operate as parallel interface ports. When this bit is 0, the above ports operate as general-purpose ports.

9.5 Interrupts of the Parallel Port

The causes of a parallel port interrupt are data reading and writing from external devices.

■ Interrupt During Parallel Port Operation

If the input buffer (PDR) is full (that is, when external data has been written) when parallel operation is enabled (RDIE bit is 1), the IBF flag is set.

If the output buffer (PDR) is empty (that is, when data has been read externally) when parallel operation is enabled (RDIE = 1), the OBF flag is cleared.

If an interrupt request enable bit is set (IRIE or IROE bit of PDCR is 1) in the above status, an interrupt request (IRQ1) is output to the CPU.

Note that the IBF and OBF bits are set or cleared according to the status of the buffer (PDR) regardless of the status of the IRIE and IROE bits.

If the IRIE or IROE bit is set (0 --> 1) to enable an interrupt when the IBF bit is 1 or the OBF bit is 0 (that is, when an interrupt cause is generated), an interrupt request is output immediately.

■ Register and Vector Table for a Parallel Port Interrupt

Table 9.5-1 Register and Vector Table for a Parallel Port Interrupt

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bits to be set		Higher	Lower
IRQ1	ILR1(007C _H)	L11(bit3)	L10(bit2)	FFF8 _H	FFF9 _H

For the operation of interrupts, see Section 3.4.2 "Processing During Interrupt Operation".

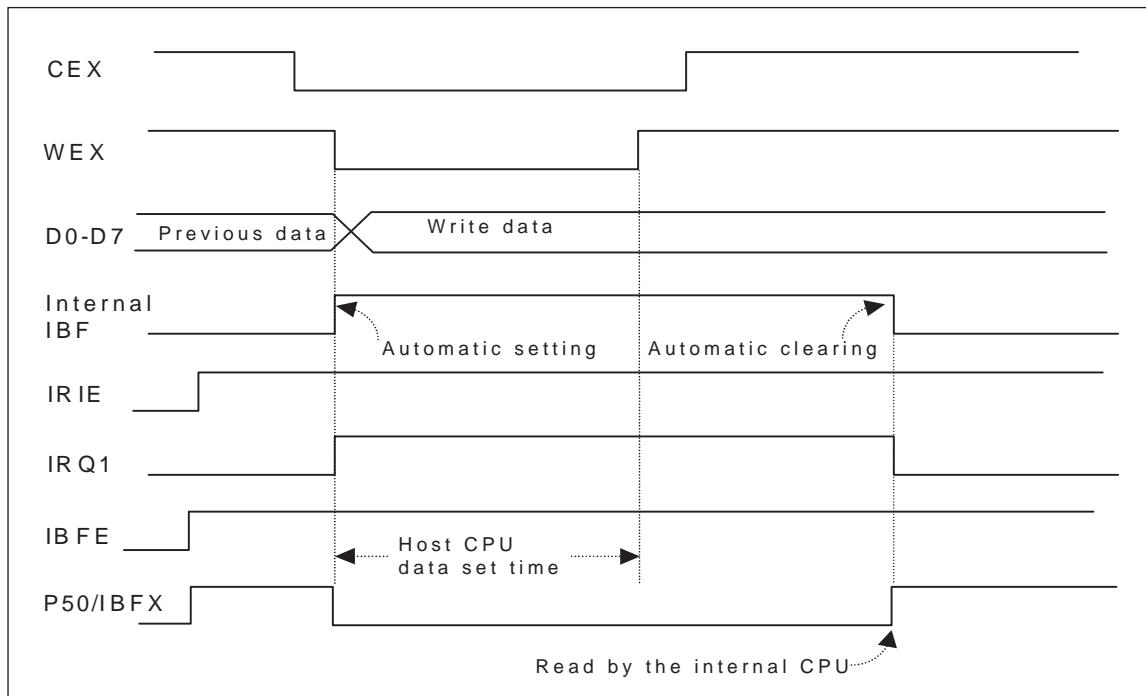
9.6 Operation of the Parallel Port

This section explains parallel port operation

■ Parallel Port Input Operation

1. When both CEX (P54) and WEX (P36) are 0, the data on external data I/O pins P40 to P47 is latched in the input buffer (PDR) on the falling edge of WEX (P36).
2. When data is latched in the input buffer (PDR), the IBF bit of the parallel port status register (PSR) is set.
3. When the IBF bit is set and the IRIE bit of the parallel port data control register (PDCR) is set, an interrupt request is output to the CPU.
4. When the content of the input buffer (PDR) is read in the CPU interrupt handling routine, the IBF flag (IBF bit of PSR), which is the interrupt cause, is cleared automatically.
5. In step 2, the status of A0 signal level is latched in the A0 bit of the parallel port status register (PSR). The latched level can be used to determine whether the content of the input buffer (PDR) is data or a command.

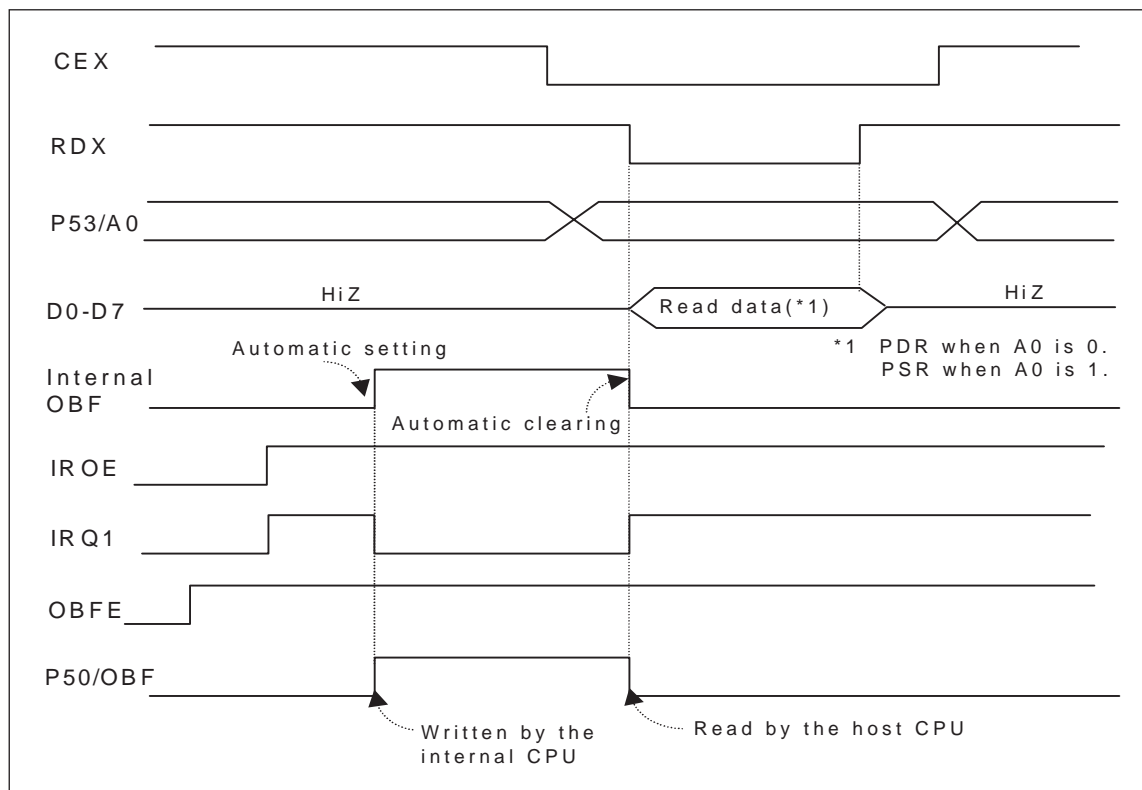
Figure 9.6-1 Parallel Port Input Operation



■ Parallel Port Output Operation

1. When output data is written to the output buffer (PDR), the OBF bit of the parallel port status register (PSR) is set.
2. When CEX (P54), PDX (P37), and A0 are 0, the data in the output buffer (PDR) is output to external data I/O pins P40 to P47 on the falling edge of PDX (P37). At the same time, the OBF bit is cleared.
3. When CEX (P54) and PDX (P37) are 0 and A0 is 1, the status in the parallel port status register (PSR) is output to external data I/O pins P40 to P47 on the falling edge of PDX (P37).
4. When OBF is cleared and the IROE bit of the parallel data control register (PDCR) is set, an interrupt request is output to the CPU.
5. When output data is written to the output buffer (PDR) in the CPU interrupt handling routine, the OBF flag (OBF bit of PSR), which is the interrupt cause, is set automatically. Then, the interrupt cause is cleared.

Figure 9.6-2 Parallel Port Output Operation



CHAPTER 10 USB FUNCTION

This chapter explains the functions and operation of the USB function circuit.

- 10.1 "Overview of the USB Function"
- 10.2 "Configuration of the USB Function Circuit"
- 10.3 "Registers of the USB Function Circuit"
- 10.4 "Interrupts of the USB Function"
- 10.5 "Functions Supported by the USB Function"
- 10.6 "Operation of the USB Function"

10.1 Overview of the USB Function

The USB function circuit is a general-purpose serial bus interface that supports the universal serial bus (USB) communication protocol. This interface supports two transmission speeds: full (12 megabits per second) and low (1.5 megabits per second).

■ Functions of the USB Function Circuit

The USB function circuit executes two-way, serial data communication with a host controller that supports the USB protocol.

- Protocol:
USB Protocol Revision 1.0 is supported.
- Transmission speed:
Full speed (12 Mbps) and low speed (1.5 Mbps) are supported.
- Device status:
A response about the status of a device is automatically sent in the USB protocol.
- Clock:
The USB interface uses a 48 MHz clock and the clock synchronization contains a digital PLL circuit. (In the function circuit, a 48 MHz clock divided by 4 is used for synchronization.)
- Data check:
Bit stripping, bit stuffing, CRC5, and CRC16 are generated and checked automatically.
- Data synchronization bit:
A data synchronization bit (DATA0/DATA1 toggle bit) is held.
- Standard USB commands:
All standard commands, except the Get/SetDescriptor and SynchFrame commands, are automatically supported. For the Get/SetDescriptor and SynchFrame commands, the status of the SETUP transaction is displayed and output as data to the CPU bus.
- Class/Vendor command:
For the Class/Vendor command, the SETUP transaction is output as data to the CPU bus.
- EndPoints:
Up to four EndPoints are supported. (EndPoint 0 is always used to control data transfer.)
- Interface/Alternate:
Switching of the interface (0 or 1) and the alternate number are controlled by software.
- Interrupt cause:
An interrupt is generated at the end of packet transfer from an EndPoints (when handshaking information to or from the USB host is received).

An interrupt is also generated for reception of a Frame Start packet, in the SETUP stage, for reception of a Set Interface command, or transition to or return from the SUSPEND stage.

- FIFO:
A FIFO interface (8 bits x 8) for transmission is built in.
- DMA transfer:
A DMA controller is built in, and internal FIFO data is transferred directly to RAM. When DMA transfer mode is specified, data can be transferred by DMA transfer to a FIFO device connected to a dedicated external I/O pin.

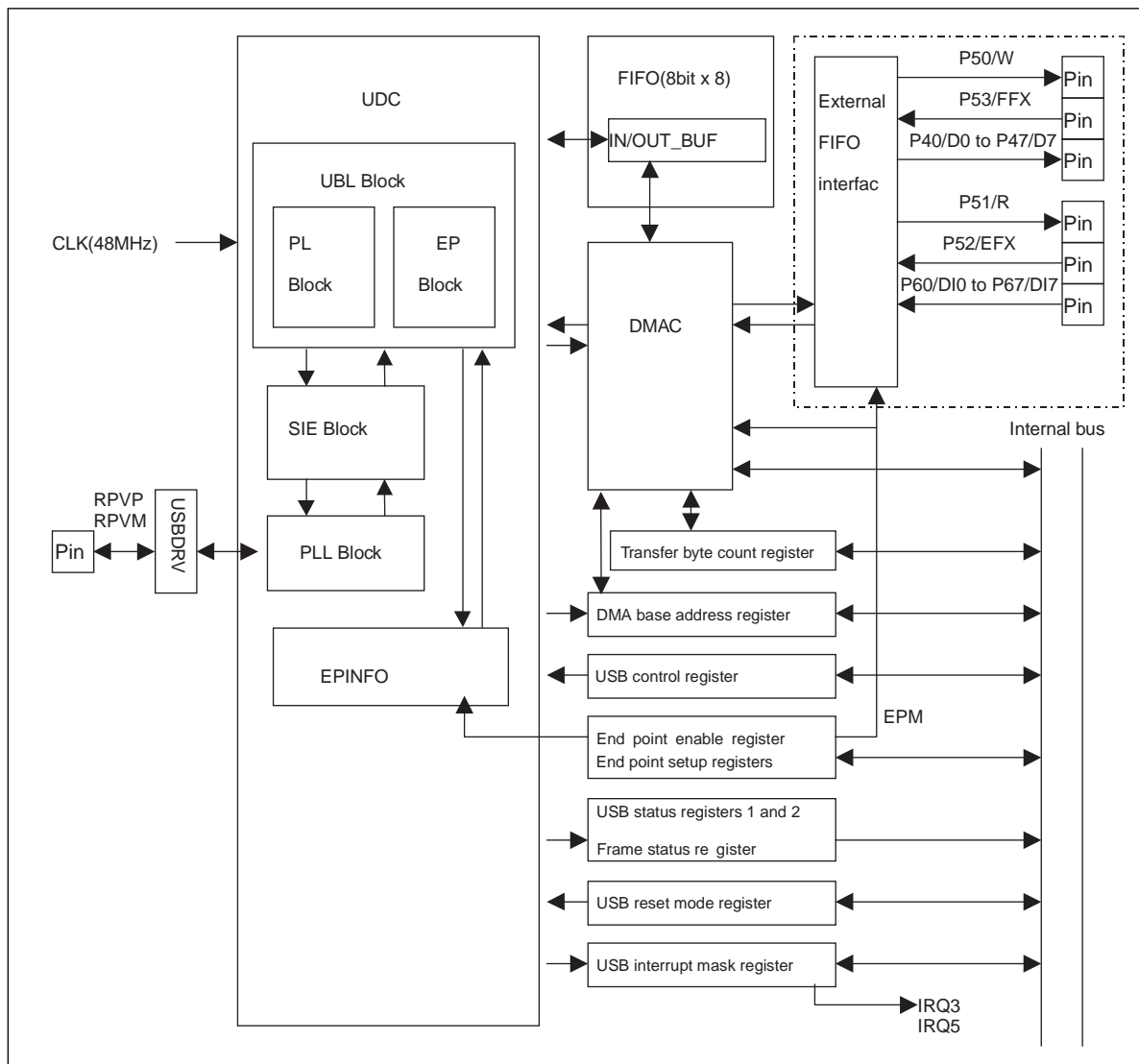
10.2 Configuration of the USB Function Circuit

The USB function circuit consists of the following four blocks:

- USB device controller (UDC)
- Transmission buffer (IN/OUT_BUF)
- DMA controller (DMAC)
- USB control registers (UMDR, DBAR, DTCR, UCTR, USTR1, USTR2, UMSKR, UFRM1, UFRM2, EPER, EPBR0, EPBRx1, and EPBRx2)

■ Block Diagram of USB Function Circuit

Figure 10.2-1 Block Diagram of USB Function Circuit



- **USB power mode register**

This register sets the power supply mode (internal supply or supply from bus) of the functions.

- **USB reset mode register**

This register specifies USB function circuit reset, operating speed (full or low), route port (host controller) connection, and resume operation.

- **DMA base address register**

This register specifies the address of the RAM area where transfer data is stored.

- **Transfer data count registers 0 to 3**

These registers indicate the byte count (specification of the number of DMA bytes to be transmitted or indication of the number of bytes received) of the transfer packets at each end point.

- **USB control register**

This register controls USB protocol data transfer.

- **USB status registers 1 and 2**

These registers indicate the termination status of USB transfer and the current information about transfer packets.

- **USB interrupt mask register**

This register masks USB transfer data termination and SOF packet reception interrupts.

- **Frame status register**

This register holds the start frame number.

- **End point enable register**

This register enables or disables the function at each end point.

- **End point setup registers**

These registers specify the configuration information (transfer type, direction, and maximum packet size) about the end points.

10.3 Registers of the USB Function Circuit

This section explains the USB function circuit registers.

■ USB Function Circuit Registers

○ USB power mode register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
PMDR 0040 _H	---	---	---	---	---	---	---	PWC	XXXXXXXX0 _B
								R/W	

○ USB reset mode register

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UMDR 0050 _H	RST	RESUM	HCON	SP	---	---	---	BFS	1000XXX0 _B
	R/W	R/W	R/W	R/W				R/W	

○ DMA base address register

[DMA base address register high for MB89589B/P589B only]

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
DBARH 004F _H	AD15	AD14	AD13	AD12	AD11	AD10	---	---	000000XX _B
	R/W	R/W	R/W	R/W	R/W	R/W			

[DMA base address register low common to MB89580B/BW series]

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
DBAR 0051 _H	---	---	AD9	AD8	AD7	AD6	AD5	AD4	XXXXXXXX _B
			R/W	R/W	R/W	R/W	R/W	R/W	

○ Transfer data count registers 0 to 3

[End points 0 and 3]

TDCR 0,3 (EndPoint 0,3)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address	0052 _H , 0057 _H	---	BC6	BC5	BC4	BC3	BC2	BC1	BC0	X0000000 _B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[End points 1 and 2]

TDCR 11,21 (EndPoint 1,2)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address 0053H,0055H		BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

TDCR 12,22 (EndPoint 1,2)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address 0054H,0056H		-----	-----	-----	-----	-----	-----	BC9	BC8	XXXXXX00 _B
								R/W	R/W	

○ USB control register

Address		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UCTR 0058H		Bfok3	Bfok2	Bfok1	Bfok0	Stall3	Stall2	Stall1	Stall0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

○ USB status register 1

Address		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
USTR1 0059H		Pkend	Setup	SOF	SETIF	BUSR	Wkup	Susp	NACK	00000000 _B
		R/W	R/W	R/W	R/W	R	R/W	R/W	R	

○ USB status register 2

Address		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
USTR2 005A H		I1AL1	I1AL0	I0AL1	I0AL0	SPK	DIR	EPC1	EPC0	XXXXXX00 _B
		R	R	R	R	R	R	R	R	

○ USB interrupt mask register

Address		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UMSKR 005B H		MPKend	MSetup	MSOF	Msetif	MBUSR	MWKUP	MSUSP	BUSRF	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

○ USB frame status register

UFRMR1		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
	Address	Frm7	Frm6	Frm5	Frm4	Frm3	Frm2	Frm1	Frm0	XXXXXXXX _B
	005C _H	R	R	R	R	R	R	R	R	

UFRMR2		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
	Address	---	---	---	---	---	Frm10	Frm9	Frm8	XXXXXXXX _B
	005D _H						R	R	R	

○ End point enable register

EPER		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
	Address	---	---	---	---	Epen3	Epen2	Epen1	Epen0	XXXX0001 _B
	005E _H					R/W	R/W	R/W	R	

○ End point (0, 1, 2, 3) setup register

[End point 0]

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR0	---	MP6	MP5	MP4	MP3	MP2	MP1	MP0	X0000000 _B
Address 005F _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[End points 1 and 2]

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR11,21	EPM	---	TYP1	TYP0	DIR1	DIR0	MP9	MP8	0X000000 _B
Address 0060 _H ,0062 _H	R/W		R/W	R/W	R/W	R/W	R/W	R/W	

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR12,22	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	00000000 _B
Address 0061 _H ,0063 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

[End point 3]

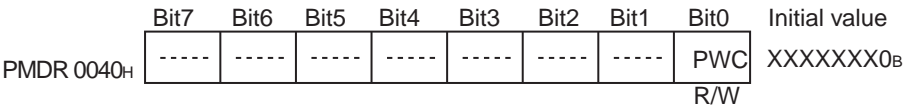
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR31	---	---	TYP1	TYP0	DIR1	DIR0	---	---	XX0000XX _B
Address 0064 _H			R/W	R/W	R/W	R/W			

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR32	---	MP6	MP5	MP4	MP3	MP2	MP1	MP0	X0000000 _B
Address 0065 _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

10.3.1 USB Power Mode Register (PMDR)

The USB power mode register sets the power supply mode (internal supply or supply from bus) of the USB functions. Bits 7 to 1 of this register are reserved. When writing data by using a byte instruction, always write 1000001 + PWC B.

■ USB Power Mode Register (PMDR)



Bits 7 to 1: Reserved

"1000001" is always written to these bits.

Bit 0: PWC (Power Control)

Bit 0 specifies the power supply mode (internal power mode or bus power mode) of the USB functions.

(This bit setting is reflected in the response of the GetStatus command.)

PWC	Power mode control bit
0	USB FUNCTION Bus power type
1	USB FUNCTION Internal power type

10.3.2 USB Reset Mode Register (UMDR)

The USB reset mode register specifies reset and other USB function circuit operations. Bits 2 to 3 of this register are reserved. When writing data to this register, always write 0 to the bits.

■ USB Reset Mode Register (UMDR)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UMDR 0050 _H	RST	RESUM	HCON	SP	---	---	RFBM	BFS	1000XX00 _B
	R/W	R/W	R/W	R/W			R/W	R/W	

Bit 7: RST (Function Reset)

Bit 7 specifies whether to reset the USB function circuit independently. In normal operation (except initialization after power-on), independent reset of the USB function circuit is not necessary. Resetting of the USB function circuit has an OR relation with system reset when power is turned on.

The initial value of this bit is 1, which indicates resetting. To cancel resetting, write 0 to this bit.

RST	Function Reset bit
0	Specifies that USB function circuit reset is canceled.
1	Specifies that the USB function circuit is reset.

Bit 6: RESUM (Resume)

When remote wakeup is enabled and the SUSP bit of status register 1 is 1 (SUSPEND mode), resuming is enabled. When 1 is written to bit 6, the SUSP bit is cleared to 0 and the resume operation is started. To specify resuming, write 1 to bit 6, wait at least two 12 MHz clock pulses (166 ns), then write 0 to bit 6 to clear it.

RESUME	USB Resume bit
0	Specifies that the USB Resume Start bit is cleared.
1	Specifies that resuming of USB operation is started.

Bit 5: HCON (USB Host Connection)

Bit 5 specifies whether to connect the USB host controller via the USB interface. When this bit is set, USB interface signals (D+ and D-) are connected.

HCON	USB Host Connection bit
0	Specifies that the USB host controller is not connected.
1	Specifies that the USB host controller is connected.

Note:

The MB89580B series specifies the connection of the host controller not by HCON but by enabling the external 1.5Ω pull-up resistor by means of general-purpose output control. Enable the pull-up resistor only after setting HCON.

Bit 4: SP (USB Speed)

Bit 4 sets the transmission speed of the USB interface. This setting can be done only once in the initialization processing after power is turned on. After setting the transmission speed, cancel USB function circuit reset with bit 7 (RST bit). The initial value of bit 4 after power-on is 0 (low speed).

SP	USB Speed bit
0	USB Low Speed (1.5Mbps)
1	USB Full Speed (12Mbps)

Bits 3 to 2: Reserved

"0" is always written to these bits.

Bit 1: RFBM (Rate FeedBack Mode)

Bit 1 selects the data toggle mode for an interrupt transfer of USB.

RFBM	Data Toggle Mode Selection bit
0	Selects the alternate data toggle mode. PID of DATA0/DATA1 is toggled only when transfer is completed normally.
1	Selects the data toggle mode. PID of DATA0/DATA1 is toggled unconditionally.

Bit 0: BFS (Buffer Size)

Bit 0 specifies the size of the buffer memory to be allocated in RAM for each end point. The specified size of buffer memory to be allocated applies to each of the four end points (end points 0 to 3).

BFS	Buffer Size bit
0	8 bytes (for each of EndPoints 0 to 3)
1	64 bytes (for each of EndPoints 0 to 3)

10.3.3 DMA Base Address Register (DBAR)

The DMA base address register specifies the DMA transfer destination address of buffer memory (mapped in RAM) for each EndPoint (0 to 3). The address specified by this register is used as the DMA transfer address for EndPoint 0. The DMA transfer addresses for EndPoint 1 to EndPoint 3 are set in a continuous area in units of the size specified by the BFS bit of the USB reset mode register.

■ DMA Base Address Register (DBAR)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
DBAR 0051 _H	---	---	AD9	AD8	AD7	AD6	AD5	AD4	XXXXXXXX _B
			R/W	R/W	R/W	R/W	R/W	R/W	

Bits 7 to 6: Reserved

"0" is always written to these bits.

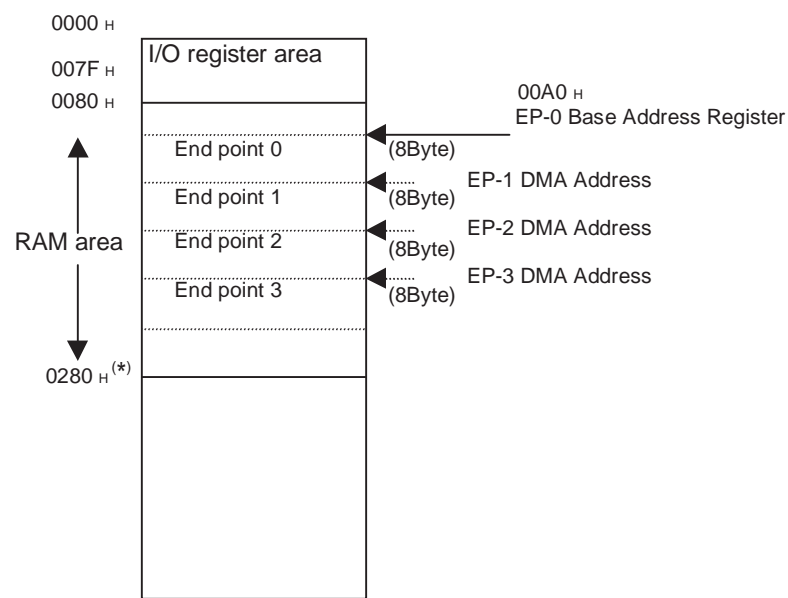
Bits 5 to 0 (DBAR): AD9 to AD4 (DMA Base Address AD9 to AD4)

Bits 5 to 0 specify the DMA transfer destination address for EndPoint 0. The DMA transfer addresses for EndPoints 1 to 3 are determined in an area beginning with the specified base address (for EndPoint 0) at intervals of the buffer size (specified by the BFS bit). Address specification format is 0XX0_H (X: any number).

The specifiable range of addresses for the DMA area is 0000_H to 03F0_H but the base address must be specified in the range of built-in RAM addresses.

(I/O area addresses 0000_H to 007F_H must be avoided.)

Example: When DBAR is 00A0_H and BFS bit of UMDR is 0 (8 bytes)



(*: 0480_H for a 1-kilobyte RAM)

10.3.4 DMA Base Address Registers for MB89589B (DBARH and DBAR)

The MB89589B/P589B uses DMA base address registers (DBARH and DBAR) to specify the DMA transfer destination address of the buffer memory (mapped in RAM) for each EndPoint (0 to 3). The address specified by using DBAR is used as the lower DMA transfer address for EndPoint 0. The lower DMA transfer addresses for EndPoint 1 to EndPoint 3 are set in a continuous area (up to 1 KB) at intervals of the size specified by the BFS bit of the USB reset mode register. DBARH specifies the destination bank address of DMA transfer.

■ DMA Base Address Register High (DBARH)

	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
DBARH	004F _H	AD15	AD14	AD13	AD12	AD11	AD10	-----	-----	000000XX _B
		R/W	R/W	R/W	R/W	R/W	R/W			

Bits 7 to 2 (DBARH): AD9 to AD4 (DMA Base Address High AD15 to AD10)

These bits specify the destination bank address of DMA transfer.

Bits 1 to 0: Reserved

"0" is always written to these bits.

■ DMA Base Address Register Low (DBAR)

	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
DBAR	0051 _H	-----	-----	AD9	AD8	AD7	AD6	AD5	AD4	XXXXXXXX _B
				R/W	R/W	R/W	R/W	R/W	R/W	

Bits 7 to 6: Reserved

"0" is always written to these bits.

Bits 5 to 0 (DBAR): AD9 to AD4 (DMA Base Address Low AD9 to AD4)

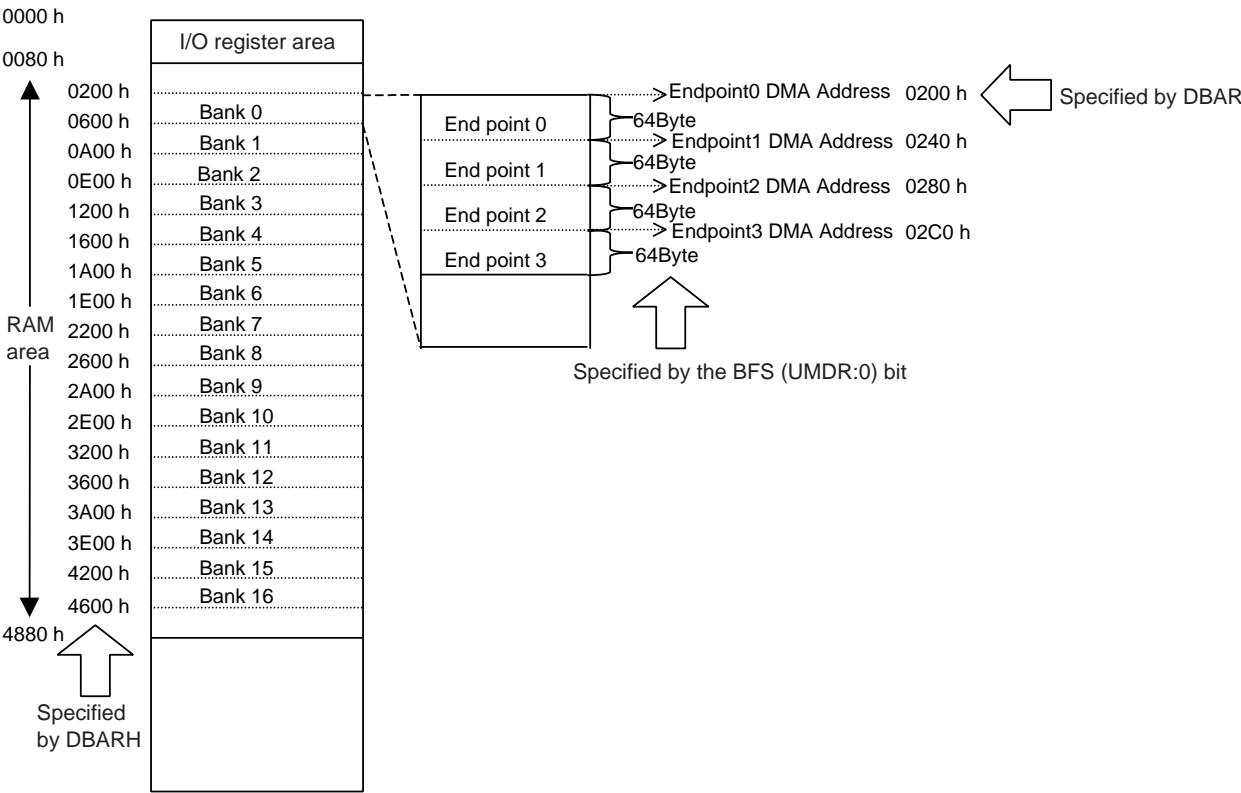
These bits specify the lower base address of the DMA transfer destination for EndPoint 0. Starting from this address, the lower addresses of DMA transfer for EndPoint 1 to EndPoint 3 are determined at intervals of the specified buffer size (specified by the BFS bit).

■ DMA Transfer Address Range

The address specified by DBARH and DBAR is XXX0H (X: arbitrary).

The specifiabale range of addresses for the DMA area is 0000H to 4880H. Specify the address within the range of internal RAM addresses. (I/O area addresses 0000H to 007FH must be avoided.)

Example) When DBARH is 00H (initial setting), DBAR is 20H, and the BFS bit of UMDR is 1 (64 bytes)



10.3.5 Transfer Data Count Registers (TDCR0 to TDCR3)

For data transfer in the OUT direction, each transfer data count register indicates the byte count of the data received in the USB protocol. For data transfer in the IN direction, each transfer data count indicates the byte count of the data to be transmitted by DMA transfer. A transfer data byte count register is available for each EndPoint (0 to 3). During data transfer in the IN direction, the count in each transfer data count register is decremented. When the count reaches 0 and the USB protocol packet transfer ends, DMA transfer ends. If the relevant interrupt enable bit is set to enable interrupts at the end of DMA transfer, an interrupt is sent to the CPU.

■ Transfer Data Count Registers (TDCR0, TDCR1, TDCR2, TDCR3)

(EndPoint 0)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR0	Address	-----	BC6	BC5	BC4	BC3	BC2	BC1	BC0	X0000000 _B
	0052 _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
(EndPoint 1)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR11	Address	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000 _B
	0053 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
(EndPoint 1)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR12	Address	-----	-----	-----	-----	-----	-----	BC9	BC8	XXXXXX00 _B
	0054 _H							R/W	R/W	
(EndPoint 2)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR21	Address	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000 _B
	0055 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
(EndPoint 2)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR22	Address	-----	-----	-----	-----	-----	-----	BC9	BC8	XXXXXX00 _B
	0056 _H							R/W	R/W	
(EndPoint 3)		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
TDCR3	Address	-----	BC6	BC5	BC4	BC3	BC2	BC1	BC0	X0000000 _B
	0057 _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

A transfer data count register is available for each EndPoint. EndPoint 0 and 3 correspond to TDCR0 and TDCR3, respectively. The maximum transfer data count for these EndPoint is 64 bytes. The maximum transfer data count for EndPoint 1 and 2 is 1,024 bytes. This maximum transfer data count is supported when the DMA transfer to external RAM is enabled by the mode specification (DMA transfer via ports 5 and 6 and support of isochronous transfer).

If the mode specified for EndPoint 1 or 2 does not support isochronous transfer (but supports the DMA transfer to internal RAM), the maximum transfer data count is 64 bytes. The same holds true for EndPoint 0 and 3.

Write 0 to all vacant bits when data is written to each transfer data count register.

TDCR0, 3 [EndPoint 0, 3]

Bits 6 to 0: BC6 to BC0 (Transfer Byte Counter)

A maximum of 64 bytes can be specified.

TDCR11, 12/21, 22 [EndPoint 1, 2]

Bits 6 to 0: BC6 to BC0 (Transfer Byte Counter)

A maximum of 1,024 bytes can be specified.

For USB OUT transfer (data reception by the USB function circuit), each transfer data count register operates as a status register indicating the byte count of the received data. The transfer data count register corresponding to each EndPoint indicates the byte count of the data the USB function circuit received from the host. This indication is done when the Pkend bit (packet transfer end status) is 1 and the DIR bit (transfer direction status) indicates the OUT direction. (The status of received-data count is valid only when the Pkend and DIR bits are 1.)

For USB IN transfer (data transmission from the USB function circuit to the host), each transfer data count register specifies the byte count of the data to be transmitted by DMA transfer. Each transfer data count register indicates the byte count for an EndPoint that supports IN direction transfer.

The maximum transfer data count for EndPoint 0 and 3 is 64 bytes. A maximum of 1,024 bytes can be transferred from an external FIFO device to EndPoint 1 and 2.

Examples:

8 bytes: 08_H

64 bytes: 40_H

10.3.6 USB Control Register (UCTR)

The USB control register controls the data transfer via the USB protocol interface.

■ USB Control Register (UCTR)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UCTR 0058 _H	Bfok3	Bfok2	Bfok1	Bfok0	Stall3	Stall2	Stall1	Stall0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bits 7 to 4: BFOK3, 2, 1, 0 (Buffer OK)

Bits 7 to 4 specify whether the transfer buffer is ready for the next transfer. Because these bits are cleared by hardware when the Pkend bit (packet transfer end status) is set (by a USB ACK or NACK response) is set, these bits must be set (to enable transfer) again before the next transfer.

BFOK3, BFOK2, BFOK1, and BFOK0 correspond to end points 3 to 0.

BFOK	Buffer Ready OK bit
0	Specifies that the transfer buffer is not ready.
1	Specifies that the transfer buffer is ready and DMA transfer is enabled.

Bits 3 to 0: STALL3, 2, 1, 0 (USB Stall Response)

When these bits are set to 1, stall status can be set for the end point (3 to 0) that is accessed.

These bits are automatically cleared when the host accesses an EndPoint in which STALL has been set.

STALL	Stall Response bit
0	Specifies that a stall response is not made.
1	Specifies that a stall response is made.

10.3.7 USB Status Register 1 (USTR1)

USB status register 1 indicates the current status of the USB interface such as the end of data transfer, progress of the setup transaction, or reception of an SOF packet. If the interrupt cause for each status is not masked, an interrupt is sent to the CPU. If the USB up-port is suspended, put the standby control register (STBC register), which is used to stop the clock, in stop mode to limit the standby mode current to less than that specified by the USB specification.

■ USB Status Register 1 (USTR1)

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address		PKen	Setup	SOF	SETIF	BUSR	Wkup	Susp	NACK	00000000 _B
USTR1	0059 _H	R/W	R/W	R/W	R/W	R	R/W	R/W	R	

Bit 7: Pkend (Transfer Packet End)

Bit 7 indicates whether the current packet transfer has ended. (This bit is set to 1 when ACK or NACK is received for transfer packets.) This bit is cleared when 0 is written to it. Writing of 1 to this bit is ignored. If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU.

Pkend	Transfer Packet End bit
0	Indicates that packets are not transferred in the USB protocol.
1	Indicates that USB protocol packet transfer has ended.

Bit 6: Setup (Setup)

Bit 6 indicates that the USB interface is in the setup stage of USB control data transfer. This bit is cleared when 0 is written to it. Writing of 1 to this bit is ignored. If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU.

Setup	Setup status bit
0	Indicates that the interface is not in the setup stage of control data transfer.
1	Indicates that the interface is in the setup stage of control data transfer.

Bit 5: SOF (Start of Frame)

Bit 5 indicates that an SOF packet has been received via the USB interface.

When this bit is 1, the frame number indicated by the USB frame status register is valid. This bit is cleared when 0 is written to it. Writing of 1 to this bit is ignored.

If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU.

SOF	Start of Frame status bit
0	Indicates that no SOF packet has been received. (The SOF bit is cleared.)
1	Indicates that an SOF packet has been received.

Bit 4: Setif (Set Interface)

Bit 4 is set when a Set Interface command, which is a USB command, is received. This bit is cleared when 0 is written to it. Writing of 1 to this bit is ignored. If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU. When this bit is set, UCTR (BFOK3 to BFOK0) is cleared automatically.

Setif	Set Interface status bit
0	Indicates that no Set Interface command has been received. (Interrupt cause is cleared.)
1	Indicates that a Set Interface command has been received.

Bit 3: BUSR (USB Bus Reset)

Bit 3 indicates that the USB interface has switched to the bus reset status. This bit is automatically cleared when the USB interface returns from the bus reset status.

BUSR	USB Bus Reset status bit
0	Indicates that the USB bus is not reset.
1	Indicates that the USB bus has been reset.

Bit 2: WKUP (Wake Up)

Bit 2 indicates that the USB interface has been restored from the suspend status. This bit is set by hardware when the suspend status signal (SUSP) changes from enabled status to disabled status. Writing of 1 to this bit is ignored. If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU.

WKUP	Wake Up status bit
0	Indicates that there is no wake-up interrupt cause. (Interrupt cause is cleared.)
1	Indicates that there is a wake-up interrupt cause.

Bit 1: SUSP (USB Suspend)

Bit 1 indicates that the USB interface enters the suspend status.

The USB interface is restored from the suspend status when the Resume bit of the USB control register is set or by a remote wake-up instruction from the host controller. The interrupt cause is cleared when 0 is written to this bit. Writing of 1 to this bit is ignored.

If the relevant bit of the interrupt mask register is set to enable interrupts, an interrupt is sent to the CPU.

SUSP	Suspend status bit
0	Indicates that there is no suspend status interrupt. (Interrupt cause is cleared.)
1	Indicates that a suspend status interrupt has been made.

Bit 0: NACK (NACK)

Bit 0 indicates that a NACK response has been sent to the USB host controller because an error (e.g., CRC or FIFO overrun) was detected in the current USB protocol data transfer. The status indicated by this bit is valid when the packet transfer end status PKEND is 1.

(Note: This bit is not an interrupt cause.)

If the current data transfer ends with a NACK response, the transfer data is not assured. If a NACK response occurs, restore the original status before transfer (prepare the transfer data in the transmission buffer again and set registers to enable transfer) and wait for the next transfer request from the host controller.

NACK	NACK response status bit
0	Indicates that a normal end response (ACK) has been sent for the current transfer.
1	Indicates that NACK has been sent for the current transfer because an error was detected.

10.3.8 USB Status Register 2 (USTR2)

USB status register 2 indicates the status (transfer direction, short-packet flag, and end point number) at the end of transfer via the USB interface. In other words, it indicates the end point attributes at the end of the USB transfer.

■ USB Status Register 2 (USTR2)

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address		I0AL0	I0AL1	I1AL0	I1AL1	SPK	DIR	EPC1	EPC0	XXXXXX00 _B
USTR2	005A _H	R	R	R	R	R	R	R	R	

Bits 7 and 6: I0AL1, I0AL0 (Interface 0 Alternate 1, 0 Number)

Bits 7 and 6 indicate the alternate number currently set for interface 0. The indicated alternate number is the one set by the Set Interface command from the USB host. Software can check whether the alternate number set by the host matches the currently managed alternate number.

I0AL1,I0AL0	Current alternate number for interface 0
0,0	Alternate 0
0,1	Alternate 1
1,0	Alternate 2
1,1	Alternate 3

Bits 5 and 4: I1AL1, I1AL0 (Interface 1 Alternate 1, 0 Number)

Bits 5 and 4 indicate the alternate number currently set for interface 1. The indicated alternate number is the one set by the Set Interface command from the USB host. Software can check whether the alternate number set by the host matches the currently managed alternate number.

I1AL1,I1AL0	Current alternate number for interface 1
0,0	Alternate 0
0,1	Alternate 1
1,0	Alternate 2
1,1	Alternate 3

Bit 3: SPK (Short Packet Frame)

Bit 3 indicates whether the data byte count of the packet to be transferred in the USB protocol is less than the maximum packet size specified for the relevant end point buffer (0 to 3). The status indicated by this bit is valid when the Pkend bit (packet transfer end status) is 1. (Note: This bit is not an interrupt cause.)

SPK	Short Packet Frame status bit
0	Indicates that the transfer packet is not less than the maximum packet size.
1	Indicates that the transfer packet is less than the maximum packet size.

Bit 2: DIR (Direction)

Bit 2 indicates the packet direction at the end of packet transfer. The status indicated by this bit is valid when the Pkend bit (packet transfer end status) is 1.

DIR	Direction status bit
0	Indicates that the packet direction is USB OUT.
1	Indicates that the packet direction is USB IN.

Bits 1 and 0: EPC1, 0 (End Point Number Code)

Bits 1 and 0 indicate the end point number of the current transfer packet. The status indicated by this bit is valid when the Pkend bit (packet transfer end status) is 1.

EPC 1,0	End Point Number status bit
0,0	EndPoint 0
0,1	EndPoint 1
1,0	EndPoint 2
1,1	EndPoint 3

10.3.9 USB Interrupt Mask Register (UMSKR)

The USB interrupt mask register controls masking of the interrupt cause bits (PKEND, SETUP, SOF, SETIF, WKUP, and SUSP) of the USB status register. Masking of an interrupt cause can be specified for each bit.

■ USB Interrupt Mask Register (UMSKR)

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UMSKR 005B _H	MPKend	MSetup	MSOF	Msetif	MBUSR	MWKUP	MSUSP	BUSRF	00000000 _B
	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	

Bit 7: MPKend (Mask Packet END Interruption)

Bit 7 controls masking of the interrupt caused by the Pkend status (data transfer end status).

The interrupt is masked (disabled) after the device is released from a power-on reset or a reset by operation of the reset bit of the reset control register.

MPKend	Pkend Interrupt Enable bit
0	Specifies that the Pkend interrupt is disabled (masked).
1	Specifies that the Pkend interrupt is enabled (not masked).

Bit 6: MSETUP (Mask SETUP Interruption)

Bit 6 controls masking of the interrupt caused by the Setup status (setup stage status of control data transfer).

The interrupt is masked (disabled) after the device is released from a power-on reset or a reset by operation of the reset bit of the reset control register.

MSetup	Setup Interrupt Enable bit
0	Specifies that the Setup interrupt is disabled (masked).
1	Specifies that the Setup interrupt is enabled (not masked).

Bit 5: MSOF (Mask SOF Interruption)

Bit 5 controls masking of the interrupt caused by the SOF status of the frame start synchronization signal.

The interrupt is masked (disabled) after the device is released from a power-on reset or reset by operation of the reset bit of the reset control register.

MSOF	SOF Interrupt Enable bit
0	Specifies that the SOF interrupt is disabled (masked).
1	Specifies that the SOF interrupt is enabled (not masked).

Bit 4: MSETIF (Mask Set Interface Interruption)

Bit 4 controls masking of the interrupt caused by reception of the host command Set Interface.

The interrupt is masked (disabled) after the device is released from a power-on reset or reset by operation of the reset bit of the reset control register.

MSETIF	SETIF Interrupt Enable bit
0	Specifies that the SETIF interrupt is disabled (masked).
1	Specifies that the SETIF interrupt is enabled (not masked).

Bit 3: MBUSR (Mask USB Bus Reset)

Bit 3 controls masking of the interrupt caused by the BUSR status of the USB bus reset status display signal. The interrupt is masked (disabled) after the device is released from a power-on reset or reset by operation of the reset bit of the reset control register. If the system has switched to the interrupt processing routine by setting this bit to enable the interrupt and detecting an USB bus reset, this bit is cleared instead of the interrupt request flag bit being cleared.

MBUSR	BUSR Interrupt Enable bit
0	Specifies that the BUSR interrupt is disabled (masked).
1	Specifies that the BUSR interrupt is enabled (not masked).

Bit 2: MWKUP (Mask Wakeup Interruption)

Bit 2 controls masking of the interrupt caused by the WKUP status of the suspend reset signal.

The interrupt is masked (disabled) after the device is released from a power-on reset or reset by operation of the reset bit of the reset control register.

MWKUP	WKUP Interrupt Enable bit
0	Specifies that the WKUP interrupt is disabled (masked).
1	Specifies that the WKUP interrupt is enabled (not masked).

Bit 1: MSUSP (Mask Suspend Interruption)

Bit 1 controls masking of the interrupt caused by the SUSP status of the suspend status indication signal.

The interrupt is masked (disabled) after the device is released from a power-on reset or reset by operation of the reset bit of the reset control register.

MSUSP	SUSP Interrupt Enable bit
0	Specifies that the SUSP interrupt is disabled (masked).
1	Specifies that the SUSP interrupt is enabled (not masked).

Bit 0: BUSRF (BUS Reset Flag)

Bit 0 indicates that resetting the USB bus has started. Writing 1 to this bit does not change the value of this bit. If an interrupt has been enabled by using a bit that corresponds to the interrupt enable register, an interrupt notification is sent to the CPU. This bit is set to 1 when the USB bus reset waveform rise is detected.

BUSRF	USB BUS Reset Flag bit
0	No interrupt cause for USB bus reset exists. Clears the interrupt cause.
1	Interrupt cause for USB bus reset exists.

10.3.10 USB Frame Status Register (UFRMR)

The USB frame status register indicates the frame number of the currently received frame synchronization signal.

■ USB Frame Status Register (UFRMR)

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UFRMR1	Address	Frm7	Frm6	Frm5	Frm4	Frm3	Frm2	Frm1	Frm0	XXXXXXXX _B
		R	R	R	R	R	R	R	R	
	005C _H									

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
UFRMR2	Address	---	---	---	---	---	Frm10	Frm9	Frm8	XXXXXXXX _B
							R	R	R	
	005D _H									

Bits 7 to 0 (UFRMR1), bits 2 to 0 (UFRMR2): FRM10 to 0 (Frame Number)

These bits indicate the current frame number of the frame synchronization packet. The frame number is valid when the SOF bit of the USB status register 1 is 1.

Bits 7 to 3 (UFRMR2): Reserved

"0" is always written to these bits.

10.3.11 EndPoint Enable Register (EPER)

The EndPoint enable register specifies whether to enable the function of each EndPoint. Each EndPoint the USB function circuit uses can be enabled or disabled.

■ EndPoint Enable Register (EPER)

The EndPoint enable register specifies whether to enable the function of EndPoints 0 to 3. Each EndPoint can be enabled or disabled.

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPER 005E _H	---	---	---	---	Epen3	Epen2	Epen1	Epen0	XXXX0001 _B
					R/W	R/W	R/W	R	

Bits 7 to 4: Reserved

"0" is always written to these bits.

Bits 3, 2, 1, 0 (Epen 3, 2, 1, 0): (EndPoint Enable 3, 2, 1, 0)

Bits 3, 2, and 1 enable or disable the function of EndPoint 3, 2, and 1. As the default setting, EndPoint 0 is enabled when power is turned on. (The EndPoint Enable 0 bit is always 1.)

The numbers attached to the bit names correspond to the EndPoint numbers.

When one of these bits is set, the EndPoint having the corresponding configuration information (Interface, Type, Dir, Max Packet Size set in the end point setup register) is enabled.

EPEN 3, 2, 1.

EPEN _x	EndPoint 3,2,1 Enable bit
0	Disables EndPoints 3, 2, and 1.
1	Enables the function of the specified EndPoint (3, 2, or 1).

10.3.12 EndPoint Setup Registers (EPBR0, EPBRx1, EPBR x2)

The EndPoint 0 setup register specifies the configuration information (EndPoint buffer) about EndPoint 0. As the default setting, the configuration of EndPoint 0 is fixed to "Control_IN" and "Control_OUT" and is included in every interface and alternate configuration.

The EndPoint setup registers x1 and x2 specify the configuration information about EndPoint 1, 2, and 3. The configuration information to be specified includes transfer type, transfer direction, and maximum packet size.

■ EndPoint Setup Registers (EPBR0, EPBRn1, EPBRn2)

○ EndPoint 0 setup register (EPBR0)

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address		---	MP6	MP5	MP4	MP3	MP2	MP1	MP0	X0000000 _B
EPBR 0	005F _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit 7: Reserved

"0" is always written to this bit.

Bits 6 to 0: MP6 to MP0 (Max Packet Size)

Bits 6 to 0 specify the maximum packet size or the maximum transfer byte count. The maximum packet size for EndPoint 0 must be 64 bytes or less.

(Also, the maximum packet size for EndPoint 0 must not be 0 bytes [bits 6 to 0 must not be all 0s].)

Example:

MP5 to MP0:

40H --> 64 bytes (maximum limit)

08H --> 8 bytes

○ End point 1 setup register (EPBR11, EPBR12)

The EndPoint 1 setup register specifies the configuration information about EndPoint 1.

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address EPBR 11 0060 _H		EPM	---	TYP1	TYP0	DIR1	DIR0	MP9	MP8	0X000000 _B
		R/W		R/W	R/W	R/W	R/W	R/W	R/W	

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address EPBR 12 0061 _H		MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit 7 (EPBR11): EPM (End Point 1 Mode)

Bit 7 specifies the DMA transfer destination for the packet transfer at end point 1.

More specifically, this bit specifies DMA transfer to an internal RAM area or DMA transfer to an external FIFO device connected to a general-purpose port. DMA transfer to internal RAM supports the maximum packet size of 64 bytes based on the buffer size allocated to the end point. Even when isochronous or bulk transfer supports a packet size larger than 64 bytes, transfer packet data can be transferred to an external FIFO port by DMA transfer provided the relevant FIFO memory is connected to a dedicated external FIFO pin.

If the EPM bit is set, the relevant port operates as a FIFO connection port regardless of the transfer direction set by DDR4 and the I/O function of general-purpose port 4 is disabled.

(When EPM = 0)

The packet data to be transferred in the USB protocol is transferred to an internal RAM area by DMA transfer.

For the transfer to internal RAM, the maximum packet size is the one (8 or 64 bytes) specified by the BFS bit (DMA buffer size) of the USB reset mode register.

(When EPM = 1)

The packet data to be transferred in the USB protocol is transferred by DMA transfer to a FIFO memory device connected to a dedicated external FIFO pin. In this mode, the maximum packet size can be 1,023 bytes or less (which is specified by MP9 to MP0).

EPM	EndPoint 1 DMA mode
0	Specifies that packet data at EndPoint 1 is transferred to an internal RAM area by DMA transfer.
1	Specifies that packet data at EndPoint 1 is transferred by DMA transfer to a FIFO device connected to a dedicated external FIFO pin.

Bit 6: Reserved

"0" is always written to this bit.

Bits 5 and 4 (EPBR11): TYP1, TYP0 (EndPoint 1 Type)

Bits 5 and 4 specify the transfer type supported by EndPoint 1.

TYP1	TYP0	EndPoint 1 Type specification
0	0	Interrupt transfer
0	1	Bulk transfer
1	0	Isochronous transfer
1	1	(This specification is not allowed.)

Bits 3 and 2 (EPBR11): DIR1, DIR0 (EndPoint 1 Direction)

Bits 3 and 2 specify the transfer direction supported by EndPoint 1.

DIR1	DIR0	EndPoint 1 Direction specification
0	0	OUT EndPoint
0	1	IN EndPoint
1	0	Both OUT and IN
1	1	(This specification is not allowed.)

Bits 1 and 0 (EPBR11) and bits 7 to 0 (EPBR12): MP9 to MP0 (Max Packet Size)

These bits specify the maximum packet size supported by EndPoint 1. When an internal RAM area is used (DMA transfer to internal RAM is specified [EPM is 0]), the maximum packet size that can be specified for EndPoint 1 is 64 bytes. When external FIFO is used (EPM = 1), the maximum packet size is 1,023 bytes. (The maximum packet size must not be 0 bytes [these bits must not be all 0s].)

Examples:

0, MP9 to MP0:

040_H --> 64 bytes (maximum limit when EPM = 0)

3FF_H --> 1,023 bytes (maximum limit when EPM = 1)

○ EndPoint 2 setup register (EPBR21, EPBR22)

The EndPoint 2 setup register specifies the configuration information about EndPoint 2. The contents of specification are the same as those of the EndPoint 1 setup register.

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR 21	Address	EPM	---	TYP1	TYP0	DIR1	DIR0	MP9	MP8	0X000000 _B
		R/W		R/W	R/W	R/W	R/W	R/W	R/W	
	0062 _H									

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
EPBR 22	Address	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	00000000 _B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	0063 _H									

Bit 7 (EPBR21): EPM (End Point 2 Mode)

Bit 7 specifies the DMA transfer destination for the packet transfer at end point 2.

Specifically, this bit specifies DMA transfer to an internal RAM area or DMA transfer to an external FIFO device connected to a general-purpose port.

If the EPM bit is set, the relevant port operates as a FIFO connection port regardless of the transfer direction set by DDR4 and the I/O function of general-purpose port 4 is disabled.

(When EPM = 0)

The packet data to be transferred in the USB protocol is transferred by DMA transfer to an internal RAM area.

For the transfer to internal RAM, the maximum packet size is the one (8 or 64 bytes) specified by the BFS bit (DMA buffer size) of the USB reset mode register.

(When EPM = 1)

The packet data to be transferred in the USB protocol is transferred by DMA transfer to a FIFO memory device connected to a dedicated external FIFO pin. In this mode, the maximum packet size can be 1,023 bytes or less (which is specified by MP9 to MP0).

EPM	End Point 2 DMA mode
0	Specifies that packet data at end point 2 is transferred by DMA transfer to an internal RAM area.
1	Specifies that packet data at end point 2 is transferred to a FIFO device connected to a dedicated external FIFO pin by DMA transfer.

Bit 6: Reserved

"0" is always written to this bit.

Bits 5 and 4 (EPBR21): TYP1, TYP0 (End Point 2 Type)

Bits 5 and 4 specify the transfer type supported by end point 2.

TYP1	TYP0	End Point 2 Type specification
0	0	Interrupt transfer
0	1	Bulk transfer
1	0	Isochronous transfer
1	1	(This specification is not allowed.)

Bits 3 and 2 (EPBR21): DIR1, DIR0 (End Point 2 Direction)

Bits 3 and 2 specify the transfer direction supported by end point 2.

DIR1	DIR0	End Point 2 Direction specification
0	0	OUT EndPoint
0	1	IN EndPoint
1	0	Both OUT and IN
1	1	(This specification is not allowed.)

Bits 1 and 0 (EPBR21) and bits 7 to 0 (EPBR22): MP9 to MP0 (Max Packet Size)

These bits specify the maximum packet size supported by end point 2. When an internal RAM area is used (DMA transfer to internal RAM is specified [EPM is 0]), the maximum packet size that can be specified for end point 2 is 64 bytes. When external FIFO is used (EPM = 1), the maximum packet size is 1,023 bytes. (The maximum packet size must not be 0 bytes [these bits must not be all 0s].)

Examples:

MP9 to MP0:

040_H --> 64 bytes (maximum limit when EPM = 0)

3FF_H --> 1,023 bytes (maximum limit when EPM = 1)

○ End point 3 setup register (EPBR31, EPBR32)

The end point 3 setup register specifies the configuration information about end point 3. End point 3 supports only DAM transfer of packet data to an internal RAM area. The maximum packet size of 64 bytes.

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address		---	---	TYP1	TYP0	DIR1	DIR0	---	---	XX0000XX _B
EPBR 31	0064 _H			R/W	R/W	R/W	R/W			

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Initial value
Address		---	MP6	MP5	MP4	MP3	MP2	MP1	MP0	X0000000 _B
EPBR 32	0065 _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bits 7 to 6 (EPBR31): Reserved

"0" is always written to these bits.

Bits 5 and 4 (EPBR31): TYP1, TYP0 (End Point 3 Type)

Bits 5 and 4 specify the transfer type supported by end point 3.

TYP1	TYP0	End Point 3 Type specification
0	0	Interrupt transfer
0	1	Bulk transfer
1	0	Isochronous transfer
1	1	(This specification is not allowed.)

Bits 3 and 2 (EPBR31): DIR1, DIR0 (End Point 3 Direction)

Bits 3 and 2 specify the transfer direction supported by end point 3.

DIR1	DIR0	End Point 3 Direction specification
0	0	OUT EndPoint
0	1	IN EndPoint
1	0	Both OUT and IN
1	1	(This specification is not allowed.)

Bits 1 to 0 (EPBR31): Reserved

"0" is always written to these bits.

Bit 7 (EPBR32): Reserved

"0" is always written to this bit.

Bits 6 to 0 (EPBR32): MP6 to MP0 (Max Packet Size)

These bits specify the maximum packet size supported by end point 3. The DMA transfer destination supported by end point 3 is internal RAM only. The maximum packet size that can be specified is 64 bytes or less. (The maximum packet size must not be 0 bytes [these bits must not be all 0s]. It must not exceed 64 bytes.)

Examples:

MP6 (MSB) to MP0 (LSB) $_H$: 08 $_H$ --> 128 bytes

40 $_H$ --> 64 bytes (maximum limit)

10.4 Interrupts of the USB Function

A USB interrupt is generated at the end of data reception from or data transmission to the USB port in the setup stage of control data transfer. An interrupt is also generated when the Start of Frame signal or Set Interface command is received or the status of the USB port is changed to or restored from the suspend status by the host or when the USB port bus reset is started. Restoration from the suspend status causes an interrupt request even in the stop status, and it is also a stop status reset cause.

■ Interrupt during USB Function Operation

When the current packet transfer ends during USB transfer, the transfer end flag bit (PKEND bit of USTR1) is set. If at that time the interrupt request enable bit is set (MPKEND bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ3) is issued to the CPU. Clearing of the interrupt request requires that the PKEND bit be cleared in the interrupt handling routine.

When the current transfer is in the setup stage during USB transfer, the interrupt request flag bit (SETUP bit of USTR1) is set. If at that time the interrupt request enable bit is set (MSETUP bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ3) is issued to the CPU. Clearing the interrupt request requires that the SETUP bit be cleared in the interrupt handling routine.

When an SOF packet is received from the host during USB transfer, the interrupt request flag bit (SOF bit of USTR1) is set. If at that time the interrupt request enable bit is set (MSOF bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ3) is issued to the CPU. Clearing the interrupt request requires that the SOF bit be cleared in the interrupt handling routine.

When a Set Interface command is received from the host during USB transfer, the interrupt request flag bit (SETIF bit of USTR1) is set. If at that time the interrupt request enable bit is set (MSETIF bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ3) is issued to the CPU. Clearing the interrupt request requires that the SETIF bit be cleared in the interrupt handling routine.

When an USB bus reset is detected, the interrupt request flag bit (BUSRF bit of UMSKR) is set to 1. In this case, if the interrupt request enable bit is set to enable (MBUSR bit of UMSKR is set to 1), the interrupt request (IRQ3) is issued to the CPU. Use the interrupt handling routine to set the interrupt request flag bit BUSRF to 0 to clear this interrupt request.

When the USB UP port has been in the idle state continuously for 3 ms or more, the port enters the suspend status and the interrupt request flag bit (SUSP bit of USTR1) is set. If at that time the interrupt request enable bit is set (MSUSPF bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ3) is issued to the CPU. Clearing the interrupt request requires that the SUSP bit be cleared in the interrupt handling routine.

When shifting to the resume status by a request from the USB UP port is detected while the USB function is in the suspend status, the relevant interrupt request flag bit (WKUP bit of USTR1) is set.

If at that time the interrupt request enable bit is set (MWKUP bit of UMSKR is 1), enabling interrupts, an interrupt request (IRQ5) is issued to the CPU. Clearing the interrupt request requires that the WKUP bit be cleared in the interrupt handling routine. The interrupt request (IRQ5) can be issued even when the CPU is in the stop status.

The PKEND, SETUP, SOF, SETIF, BUSRF, SUSP, or WKUP bit is set when the relevant interrupt cause is detected regardless of the values of the MPKEND, MSETUP, MSOF, MSETIF, MBUSR, MSUSP, and MWKUP bits.

Reference:

When the PKEND, SETUP, SOF, SETIF, BUSRF, SUSP, or WKUP bit is 1, an interrupt request is issued immediately after the MPKEND, MSETUP, MSOF, MSETIF, MBUSR, MSUSP, or MWKUP bit, respectively, is set (from 0 --> 1).

■ **Registers and Vector Tables Related to USB Interrupts**

Table 10.4-1 Registers and Vector Tables Related to USB Interrupts

Interrupt name	Interrupt level setting register			Vector table address	
	Register	Bit to be set		Higher	Lower
IRQ3	ILR1(007C _H)	L31(bit7)	L30(bit6)	FFF4 _H	FFF5 _H
IRQ5	ILR2(007D _H)	L51(bit3)	L50(bit2)	FFF0 _H	FFF1 _H

For the operation of interrupts, see Section 3.4.2 "Processing During Interrupt Operation".

10.5 Functions Supported by the USB Function

This section explains the end point setup and DMA transfer functions that are supported by the USB.

■ Functions Supported by the USB

The USB supports the following functions:

- **Config: 1**

One device configuration is supported.

- **Interface: 1, 0**

Two interfaces are supported as devices.

- **Alternate: 0, 1, 2, 3**

Up to four alternate devices are supported. The alternate devices set in a device configuration are managed by software. The alternate attribute of each specified end point is not checked by hardware. The current alternate attribute of each end point is managed by software. If the alternate device configuration is changed by a standard device request or the Set Interface command, the end point configuration definition can be changed to update the alternate attribute.

- **End point: 0, 1, 2, and 3**

The USB supports four end points (0, 1, 2, and 3).

End point 0:

End point 0 is used for the default pipe for a USB device. End point 0 supports only control transfer. The transfer packet size supported by end point 0 is set in the end point setup register (EPBR0). The maximum transfer packet size that can be specified is 64 bytes.

End point 1:

Characteristics of end point 1 are set in the end point setup register (EPBR1). The items set are the interface number (0 or 1) of end point 1, supported transfer type (interrupt, bulk, or isochronous transfer), transfer direction (IN or OUT), and maximum packet size (1,023 bytes). End point 1 supports data transfer to an external FIFO device. If a FIFO device is connected to a dedicated external I/O pin, transfer packet data can be transferred directly to the FIFO device by DMA transfer. For DMA transfer of packet data to internal RAM, the maximum packet size is 64 bytes.

End point 2:

Characteristics of end point 2 are set in the end point setup register (EPBR2). The items set are the interface number (0 or 1) of end point 2, supported transfer type (interrupt, bulk, or isochronous transfer), transfer direction (IN or OUT), and maximum packet size (1,023 bytes). End point 2 supports data transfer to an external FIFO device. If a FIFO device is connected to a dedicated external I/O pin, transfer packet data can be transferred directly to the FIFO device by DMA transfer. For DMA transfer of packet data to internal RAM, the maximum packet size is 64 bytes.

End point 3:

Characteristics of end point 3 are set in the end point setup register (EPBR3). The items set are the interface number (0 or 1) of end point 3, supported transfer type (interrupt, bulk, or isochronous transfer), transfer direction (IN or OUT), and maximum packet size (64 bytes). End point 3 does not support data transfer to an external FIFO device, but supports only DMA transfer of packet data to internal RAM. The maximum packet size is 64 bytes.

■ DMA Transfer to External FIFO Device

End points 1 and 2 have a DMA transfer function that transfers packet data directly to a FIFO device that is connected to a dedicated external I/O pin. This function can be specified by the EPM bit of the end point 1 setup and end point 2 setup registers.

Packet data is read from external I/O port 5 (read control signal) and port 6 (8-bit input data signal) and transferred to the USB host controller. Data reading and transfer are executed when the DMA transfer to an external FIFO device is enabled, and the transfer direction of the relevant end point is IN (transfer from a device to the USB host controller).

If the transfer direction of the end point is OUT, received data is output to the external FIFO device through the external I/O port (write control signal) and port 4 (8-bit output signal).

○ Interface for DMA transfer to external FIFO device

- Transfer in the IN direction
 - For connection of the external FIFO device, general-purpose port 6 (P60 to P67) is used for data input and P51 (R) and P52 (EFX) are used for control signals.

P67 (MSB) to P60 (LSB): Transfer data input port

P52 (EFX): Signal indicating that transfer data has been prepared in the external FIFO device. This pin is connected to the EMPTY signal pin of the external FIFO device.

Note:

The input level should be corresponded to CMOS 3.3V.

P51 (R) : Strobe signal to take in transfer data

A 167 ns pulse is output to this pin. This pin is connected to the read strobe signal pin of the external FIFO device.

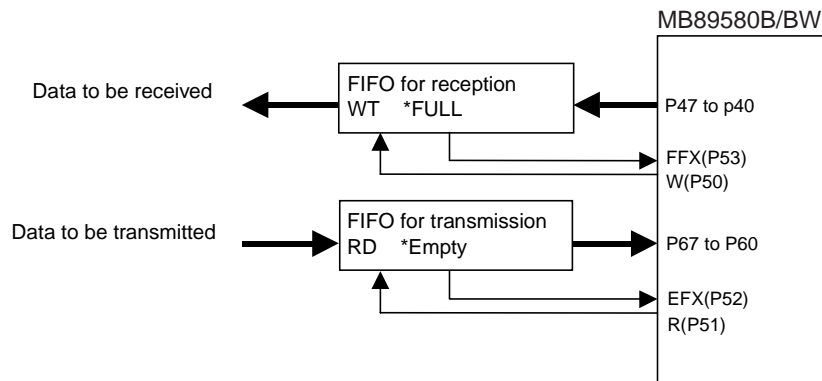
- Transfer in the OUT direction
 - For connection of the external FIFO device, general-purpose port 4 (P40 to P47) is used for data input and P50 (W) and P53 (FFX) are used for control signals.

P47 (MSB) to P40 (LSB): Transfer data output port

P53 (FFX): Signal indicating that the external FIFO device is not full and transfer data can be written to it. This pin is connected to the FULL signal pin of the external FIFO device.

P50 (W) : Strobe signal indicating that transfer data is valid

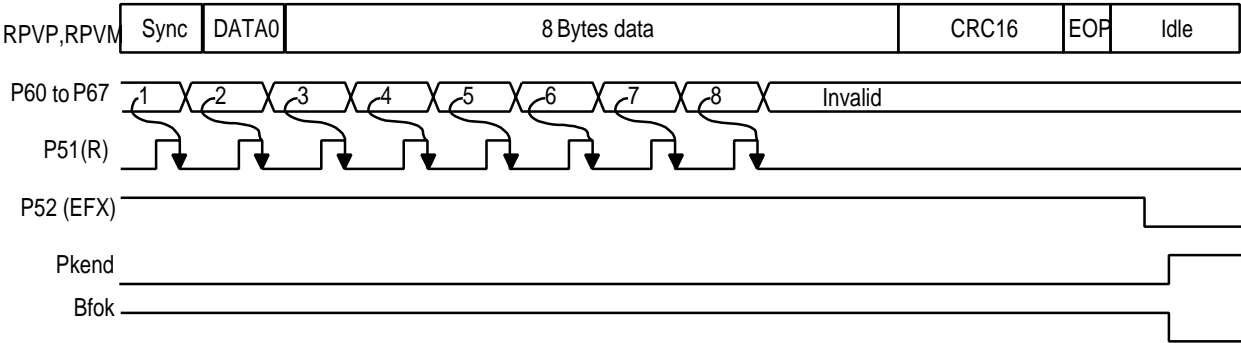
A 167 ns pulse is output to this pin. This pin is connected to the write strobe signal pin of the external FIFO device.



■ DMA Transfer to or from External FIFO

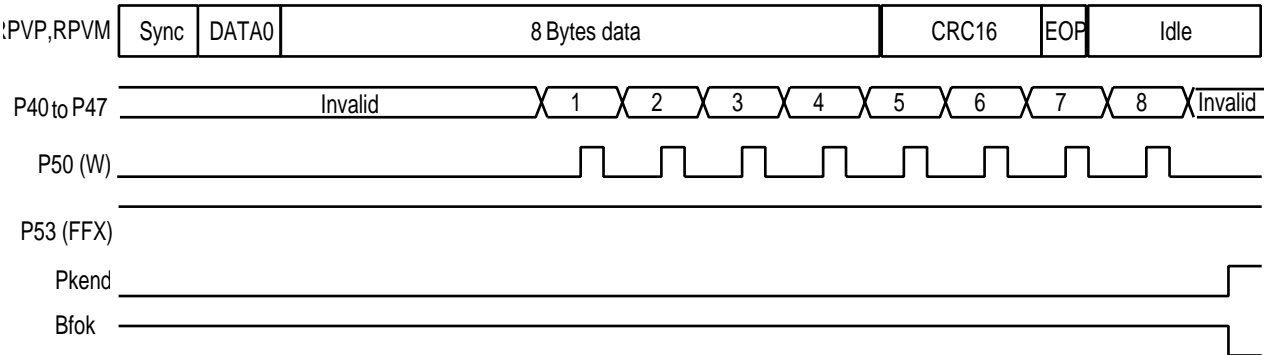
○ Transfer in the IN direction (8-byte transfer)

If the transfer direction of EndPoint is IN, data from the external FIFO is read through P60 to P67 at the falling edge of P51 (R) and transferred to the USB host controller.



○ Transfer in the OUT direction (8-byte transfer)

If the transfer direction of EndPoint is OUT, the data received from the USB is output to the external FIFO through P40 to P47 by P50 (W).

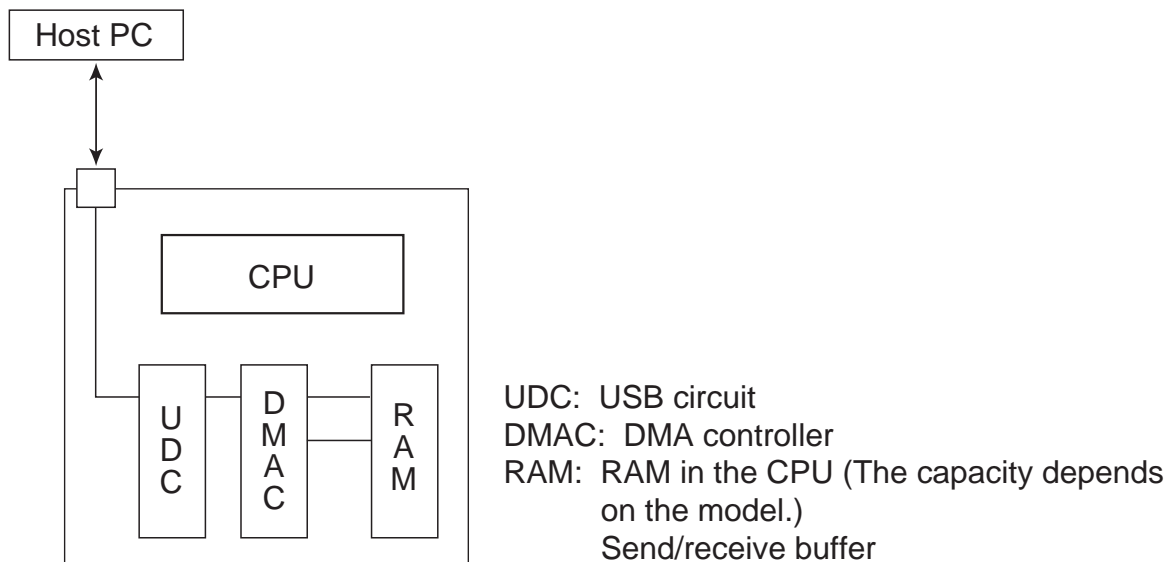


10.6 Operation of the USB Function

The USB conforms to the Universal Serial Bus (USB) communication protocol (Revision 1.0). The USB function circuit is hardware implementing the USB communication. The packet communication with the host PC does not need to be considered and the internal RAM can be accessed by DMAC.

■ Operation of the USB Function

The USB function circuit transfers packets to or from the host PC supporting the USB protocol. The host PC and a device are connected by enumeration, after which communication of the corresponding transfer type by the device driver starts.



Taking enumeration as an example, this section describes the USB communication between the host PC and a device.

To explain the entire processing, operations of the registers and USB packets are shown below:

○ Enumeration

Enumeration is the first processing to establish the connection between the host PC and a device for USB.

The host PC checks the type of the device connected on the USB bus.

The USB control transfer (USB transfer type) is used. (This is determined by using the USB specifications.)

Of the four end points, EndPoint 0 (EP0) is used. (USB specifications)

Table 10.6-1 Enumeration

Scenario	Direction of transfer	Operation overview
Connection detection	Host ← Device	A connection must be detected to start any operation.
USB bus reset	Host → Device	No processing is executed until a USB bus reset.
Descriptor acquisition	Host ← Device	Descriptor data is returned to the host.
Address setting	Host → Device	An arbitrary address is assigned from the host.
Descriptor acquisition (device)	Host ← Device	Descriptor data is returned to the host.
Descriptor acquisition (configuration)	Host ← Device	Descriptor data is returned to the host.
Configuration setting	Host → Device	A configuration number is assigned from the host.

■ Connection Detection

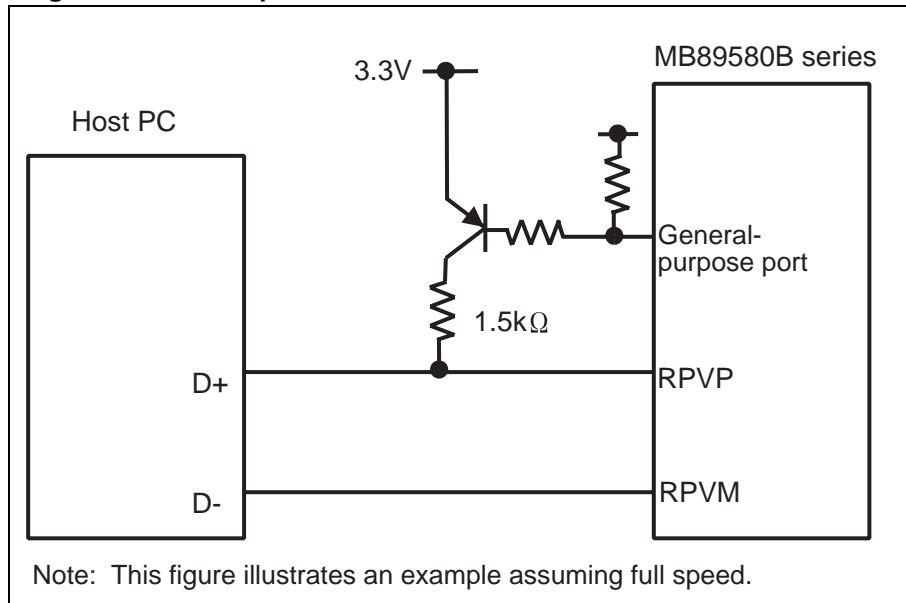
The device gives this information to the host PC.

The host monitors two signal lines (D+ and D-) of the USB bus and recognizes that the device is connected when either signal goes high level.

○ The MB89580B series requires the following processing:

1. Initial setting (initialization of all circuits including USB function registers)
2. Enable HCON of the UMDR register (connect the internal circuit to RPVP and RPVM).
3. Enable the pull-up resistor (connect the host controller) by controlling the output of a single general-purpose port with software. (RPVP and RPVM are in the high-impedance state until the USB connection is made.)

Figure 10.6-1 Sample Connection of a Model of the MB89580B Series



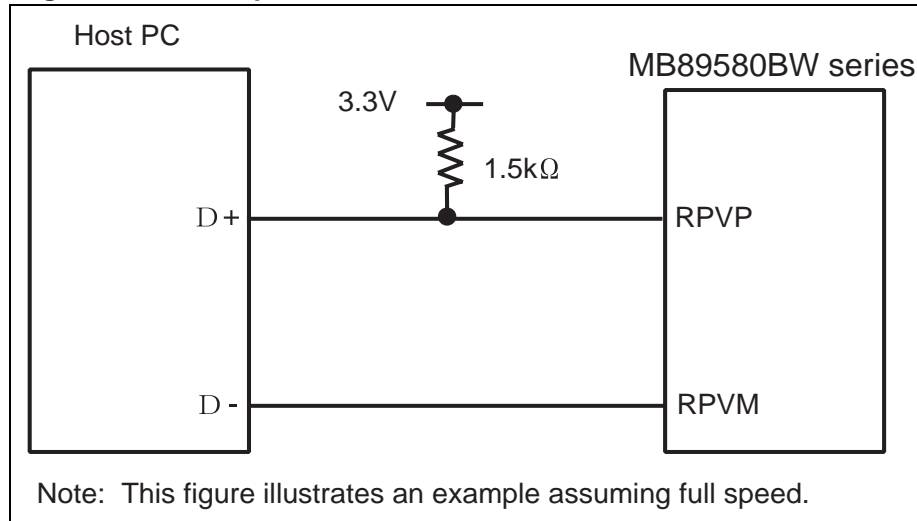
○ The MB89580BW series requires the following processing:

1. Initial setting (initialization of all circuits including USB function registers)
2. Enable HCON of the UMDR register (connect the host controller). (The outputs of RPVP and RPVM are kept low level until the USB connection is made.)

Note:

While HCON is negated, the host PC is not connected.

Figure 10.6-2 Sample Connection of a Model of the MB89580BW Series



■ USB Bus Reset

The host PC gives the information to the device.

The USB function circuit is initialized.

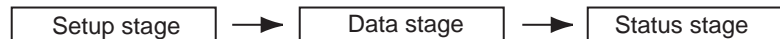
The device requires the following processing:

1. Initialize the configuration. (If two or more interface numbers are supported, restore the first number.)
2. If the send/receive buffer of the DMAC transfer destination is ready, enable BFOK0 of the UCTR register (allow DMAC transfer).

■ Descriptor Acquisition

At the request of the host PC, the device gives data to the host.

More specifically, the communication is executed in these three steps:



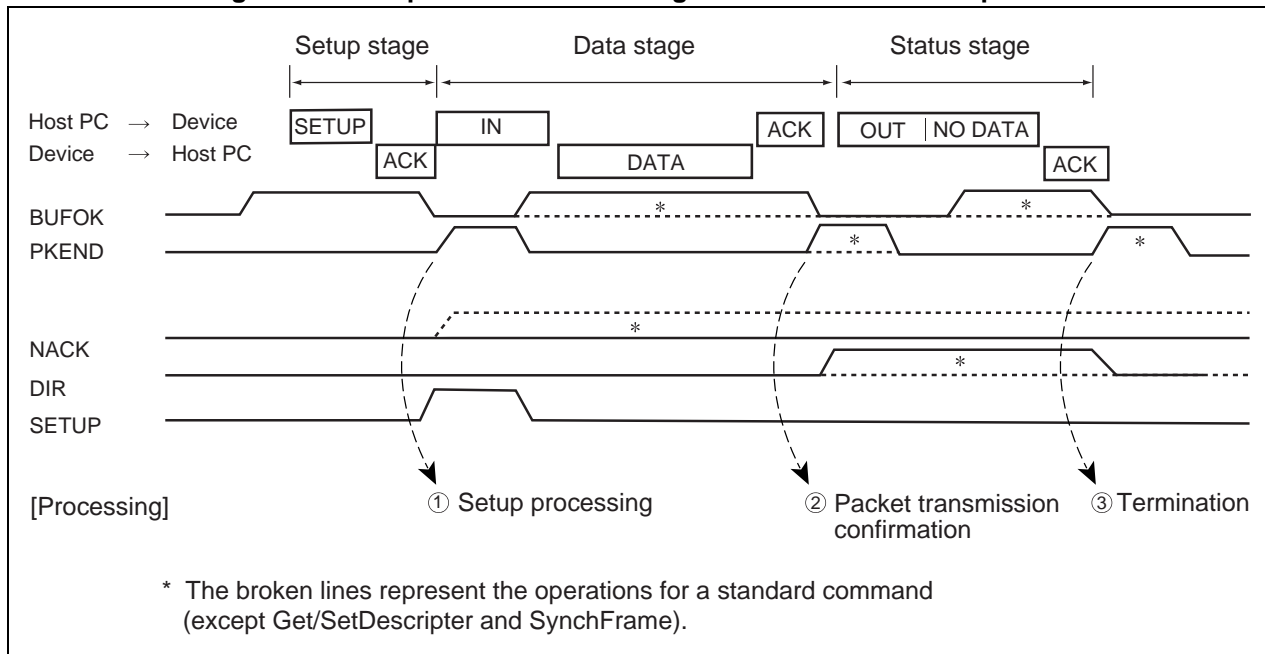
In the setup stage, the system checks whether a packet from the host PC is normally sent to the device, after which the command is decoded. In addition, the descriptor information to be returned in the data stage is prepared in the RAM area for transmission. In the data stage, the system checks whether data is normally sent from the host PC. In the status stage, the host PC transfers a no-data packet, after which the termination is executed.

10.6.1 Operations of Each Register for Command Responses

This section explains the processing method (architecture) of USB packets by carrying out the basic operations and control of the register.

■ Operations of Each Register for Command Responses

Figure 10.6-3 Operations of Each Register for Command Responses



○ Setup processing

The firmware executes processing in units of handshake sessions. This means that the processing is executed in units of stages of each packet as illustrated in Figure 10.6-3 "Operation of Registers when a Response to a Command is Made". The above figure is taken as an example for the following explanation.

BUFOK is enabled before the SETUP packet of the setup stage arrives.

Then, when the setup stage is received, BUFOK is cleared, and PKEND is set.

When PKEND is set, the system checks whether the NACK flag is cleared or SETUP is enabled for the packet, after which the command is decoded for a variety of setting processes. If the NACK flag is set for a standard command, the subsequent stage processing is not executed because a response is automatically made for a standard command. SETUP is cleared so that the next stage, the setup stage, can be recognized. Then, PKEND is cleared and BUFOK is set to receive the packet of the data stage.

○ Packet transmission confirmation

When the data stage is received, BFOK is cleared, and PKEND is set.

When PKEND is set, the system checks whether the NACK flag is cleared for the packet. If necessary, the update of the RAM data of the DMA transfer destination, DMA address change, and other processing are executed. Then, PKEND is cleared and BFOK is set in preparation for packet reception in the next stage, the status stage, or the data stage retransfer from the host.

○ End processing

If the status stage is received normally, BFOK is cleared and PKEND is set.

When the DIR value indicates the change of packet transfer direction from IN to OUT, it is confirmed that the packet is in the status stage. The end processing of the last packet is then executed.

The table below indicates the operation when each stage of an USB command is received.

Table 10.6-2 Register Changes During Command Operation

Register value					State
BUFOK	PKEND	NACK	DIR	SETUP	
0	0	0	0	0	Initial state
1	0	0	0	0	Packet wait state
0	1	1	0	1	SETUP packet received. IN packet for standard commands
0	1	0	0	1	SETUP packet received for a class vendor command other than the Get/SetDescriptor, SynchFrame, and standard commands
0	1	0	1	0	IN packet received for a class vendor command other than the GetDescriptor, SynchFrame, and standard commands
0	1	0	0	0	OUT packet received for a class vendor command other than the SetDescriptor and standard commands
0	1	0	0	0	OUT status received for a class vendor command other than the GetDescriptor, SynchFrame, and standard commands
0	1	0	1	0	IN status received for a class vendor command other than the SetDescriptor and standard commands

■ Operations when Receiving the Setup Stage of the USB Standard Command

The table below lists the operations when the setup stage of the USB standard command is received.

Table 10.6-3 Operations of Setup Stage of the Standard Command (1 / 2)

bRequest	Value	Register			Hardware operation	Firmware processing	Host PC hand-shake
		NACK	DIR	SETUP			
GetStatus	0	1	0	1	The current status is returned to the host PC. The UDC hardware automatically gives the status.	Enables BUFOK0 to receive the next packet.	ACK
ClearFeature	1	1	0	1	The function selected by using WValue is cleared on the UDC hardware.	Enables BUFOK0 to receive the next packet.	ACK
-	2	1	0	1	Unsupported	Enables BUFOK0 to receive the next packet.	ACK
SetFeature	3	1	0	1	The function selected by WValue is enabled.	Enables BUFOK0 to receive the next packet.	ACK
-	4	1	0	1	Unsupported	Enables BUFOK0 to receive the next packet.	ACK
SetAddress	5	1	0	1	A new address number is set. The UDC hardware retains the number specified by the host PC.	Enables BUFOK0 to receive the next packet.	ACK
GetDescriptor	6	0	0	1	The current descriptor data is returned to the host PC. DMAC automatically gives the contents of RAM prepared by the firmware as the data.	Stores the data of the descriptor indicated by the number of the function selector obtained by interpreting the command, in the RAM area indicated by DBAR and TDCR0. Enables BUFOK0 to receive the next packet.	ACK
SetDescriptor	7	0	0	1	The descriptor data is sent from the host. DMAC automatically writes the data into RAM.	Enables BUFOK0 to receive the next packet.	ACK
GetConfiguration	8	1	0	1	The current configuration number is returned to the host PC. The UDC hardware automatically gives the number.	Enables BUFOK0 to receive the next packet.	ACK
SetConfiguration	9	1	0	1	A new configuration number is set. The UDC hardware retains the number specified by the host PC.	Enables BUFOK0 to receive the next packet.	ACK

Table 10.6-3 Operations of Setup Stage of the Standard Command (2 / 2)

bRequest	Value	Register			Hardware operation	Firmware processing	Host PC hand-shake
		NACK	DIR	SETUP			
GetInterface	10	1	0	1	The current interface number is returned to the host PC. The UDC hardware automatically gives the number.	Enables BUFOK0 to receive the next packet.	ACK
SetInterface	11	1	0	1	The alternate setting of the specified interface is changed, and the SETIF bit is set. The alternate setting can be checked by using the USTR2 register.	The configuration of each EndPoint can be dynamically changed by managing the alternate setting for each interface. In accordance with the specified alternate setting, the processing of the amount and contents of data to be sent or received changes. Clears the SETIF bit to generate the next SetInterface command reception interrupt. Enables BUFOK0 to receive the next packet.	ACK
SynchFrame	12	0	0	1	A synchronization frame number is returned to the host. DMAC automatically gives the contents of RAM prepared by the firmware as the number.	Stores the synchronization frame number in the RAM area indicated by DBAR and TDCR0. Enables BUFOK0 to receive the next packet.	ACK

10.6.2 Suspend Function

When a USB device is not operating, the device needs to go into suspend mode and then enter the low-power consumption state. The current consumption required of a device while in suspend mode must be 500 μ A or less in the bus power supply (power supplied from the USB cable) device configuration. To implement this, the STOP mode (the main clock is stopped), which realizes the lowest power consumption of all standby modes of the CPU operation modes, must be set.

■ Suspend Processing

When the USB device core detects the suspend state, SUSP of the USR1 register is set.

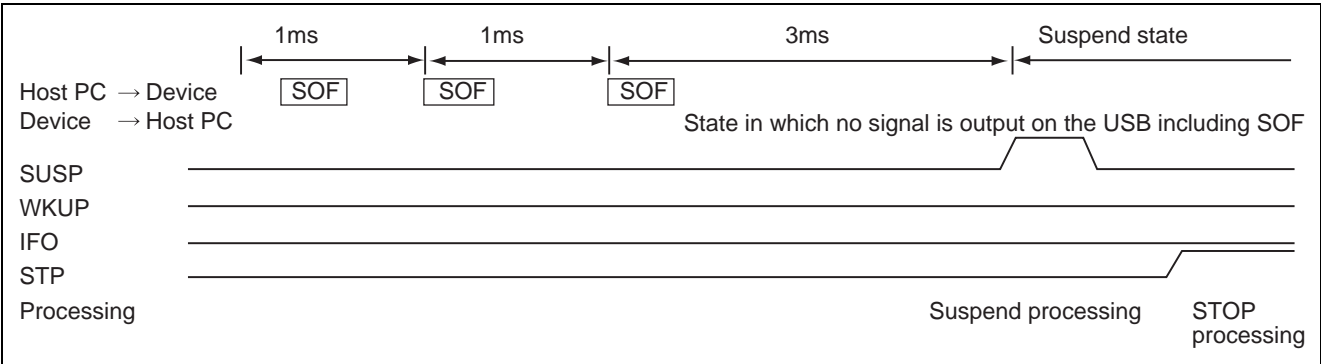
[Processing] The following processing is then required:

1. Read and clear the SUSP status.
2. If the device supports remote wake-up, wait for 2 ms (5 ms in the USB specifications, including the suspend time). (This is intended to prevent remote wake-up during this time.)
3. The following are effective for reducing power consumption in the suspend processing (some of them may not be necessary).
 - Setting the input or output of a general-purpose port
 - Adjusting the pull-up or pull-down voltage level in the user system
 - Reducing the power consumption of the user peripheral function
4. Enable external interrupt registers EIF and EIE.
5. Set STP of the STBC register for the STOP mode.

Note:

For details of the stop mode operation, see Section 3.7.3 "Stop mode".

Figure 10.6-4 Suspend Operation



■ Clearing the Suspend State

A device in the suspend state and also in STOP mode resumes operation if signals appear in at least one of the following:

1. Two USB buses (D+, D-)
2. External interrupt pins (INT0 to INT7)
3. External reset pin ($\overline{\text{RST}}$)

When the USB function circuit detects a resume signal (that is, the wake-up state), WKUP of the USTR1 register is set.

10.6.3 Wake-up Function

To change a USB device from the suspend state to the wake-up state, the USB provides the following two means:

- Remote wake-up from a device
- Wake-up from the host PC

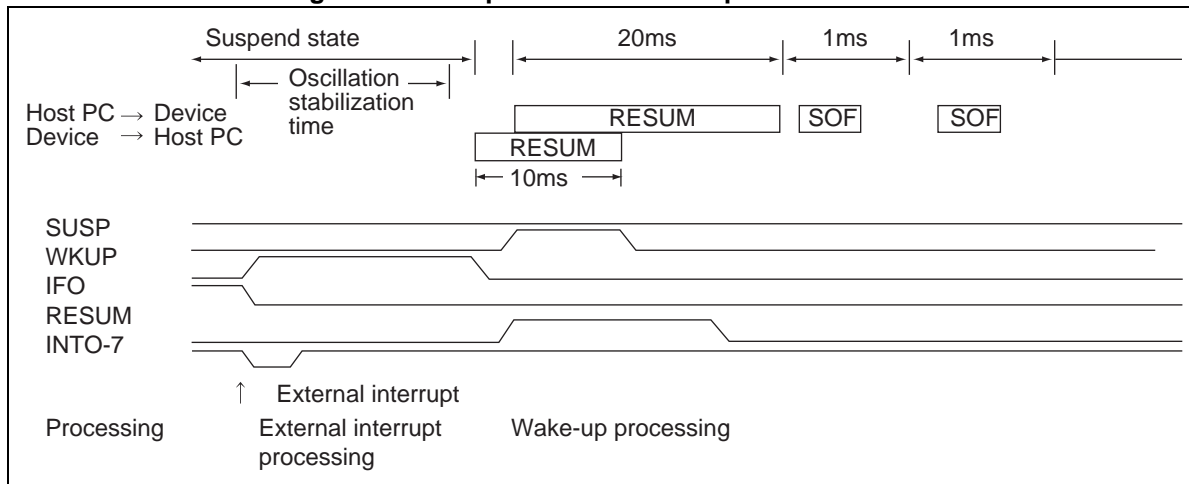
■ Remote Wake-up from a Device

The resume signal is sent from a device to the host PC.

[Processing] This necessitates the following processing:

1. Disable the EIF and EIE registers by an external interrupt if necessary.
2. Enable RESUM of UMDR.
3. Read and clear the WKUP status.
4. Restore the lower power consumption carried out in the suspend processing to its normal state.
 - Setting the input or output of a general-purpose port
 - Setting pull-up or pull-down in the user system
 - Operation of the user peripheral function with normal-power consumption

Figure 10.6-5 Operation of Wake-up from a Device



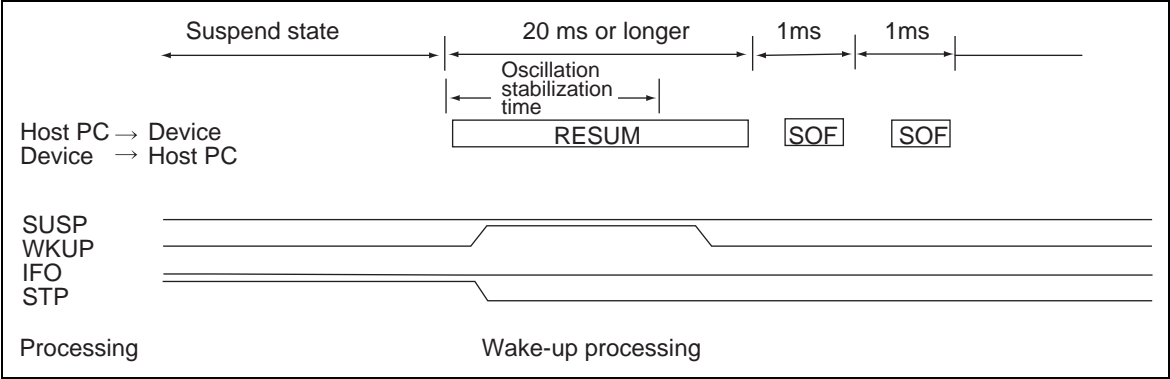
■ Wake-up from the Host PC

The resume signal is sent from the host PC to a device.

[Processing] Then, the following processing is required.

Carry out steps "3." and "4." of remote wake-up from a device.

Figure 10.6-6 Wake-up Operation from the Host



CHAPTER 11 UART/SIO

This chapter explains the functions and operation of UART/SIO.

- 11.1 "Overview of the UART/SIO"
- 11.2 "Configuration of the UART/SIO"
- 11.3 "Pins of the UART/SIO"
- 11.4 "Registers of the UART/SIO"
- 11.5 "Interrupts of the UART/SIO"
- 11.6 "Operation of the UART/SIO"
- 11.7 "Explanation of Operating Mode 0"
- 11.8 "Explanation of Operating Mode 1"

11.1 Overview of the UART/SIO

UART/SIO is the general serial data communication interface. Serial data transfer of variable data length is possible in either clock synchronous or clock asynchronous mode. The transfer format is NRZ, and the transfer rate can be set by a special baud rate generator, external clock, or internal timer.

■ UART/SIO Functions

UART/SIO sends and receives serial data (serial data input and output) to and from other CPUs and peripheral devices.

- UART/SIO has a full-duplex double buffer that enables full-duplex, bidirectional transmission.
- Synchronous transfer mode or asynchronous transfer mode can be selected.
- One of 14 baud rates provided by the internal baud rate generator can be selected. Also, the other baud rate can be set when an external clock is input.
- The data length is variable. Set 7 or 8 bits when parity is used, and 8 to 9 bits when no parity is used (Table 11.1-1 "UART/SIO Operating Modes").
- The data transfer format is NRZ (Non Return to Zero).

Table 11.1-1 UART/SIO Operating Modes

Operating mode	Data length		Synchronous mode	Stop bit length
	No parity	Parity		
0	7	8	Asynchronous	1 bit or 2 bits
	8	9		
1	8		Synchronous	-

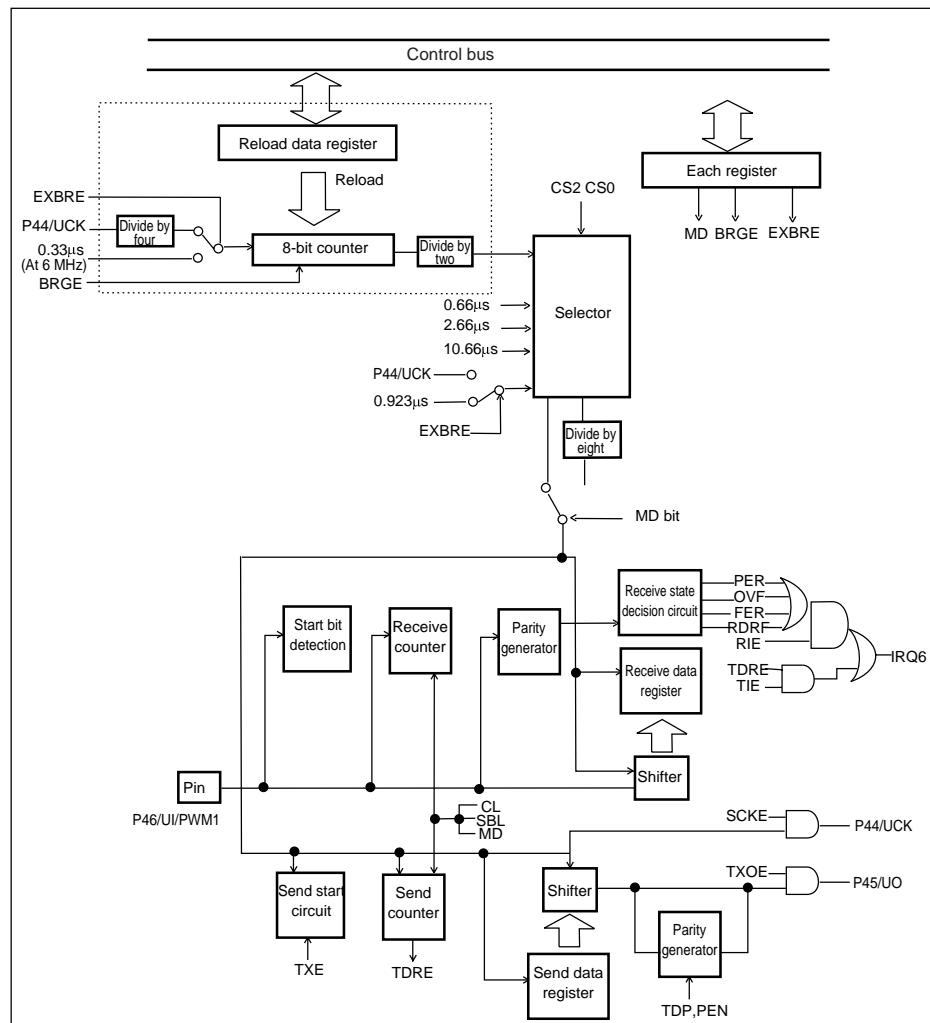
11.2 Configuration of the UART/SIO

UART/SIO consists of the following seven blocks:

- Serial clock switch register (SCS)
- Serial mode control register (SMC1)
- Serial mode control register (SMC2)
- Serial rate control register (SRC)
- Serial status and data register (SSD)
- Serial input data register (SIDR)
- Serial output data register (SODR)

■ Block Diagram of UART/SIO

Figure 11.2-1 UART/SIO Block Diagram



○ **Serial clock switch register (SCS)**

The serial clock switch register switches the input clock of the special baud rate generator to the internal clock or divide-by-four clock of the external clock. It also switches the clock selected when CS2, CS1, CS0 = 100B to the internal clock or external clock.

○ **Serial mode control register (SMC1)**

The serial mode control register controls the UART/SIO operating mode. It sets the existence of parity, length of stop bit, operating mode (data length), synchronous or asynchronous, and serial clock.

○ **Serial mode control register (SMC2)**

The serial mode control register controls the UART/SIO operating mode. It sets serial clock enable and disable, serial data output enable and disable, serial port and general-purpose port switching, and enabling and disabling of interrupts.

○ **Serial rate control register (SRC)**

The serial rate control register controls the UART/SIO data transfer speed (baud rate).

○ **Serial status and data register (SSD)**

The serial status and data register indicates the status of UART/SIO receive and send, and the error status.

○ **Serial input data register (SIDR)**

The serial input data register stores the received data. Serial data is converted and stored in this register.

○ **Serial output data register (SODR)**

The serial output data register sets the send data. The data written to this register is converted to serial and output.

11.3 Pins of the UART/SIO

This section explains the pins related to UART/SIO and contains a pin block diagram.

■ Pins Related to UART/SIO

The pins related to UART/SIO are the clock input and output pin (P44/UCK), serial data output pin (P45/UO), and serial data input pin (P46/UI). These pins can be switched by using the port selection bit (TX0E: SCKE).

P44/UCK:

Functions as a general-purpose input and output port (P44) and UART/SIO clock input and output pin (hysteresis input) (UCK). When clock output is enabled (SMC2: SCKE = 1), the pin functions as the UART/SIO clock output pin (UCK) regardless of the value of the corresponding port direction register. Do not select the external clock for this use (except when SMC1: CS2, CS1, CS0 = 100B). When the pin is used as the UART/SIO clock input pin, disable clock output (SMC2: SCKE = 0), and set it as the input port (DDR: bit 4 = 0) using the corresponding port direction register. For this use, the external clock (SMC1: CS2, CS1, CS0 = 100B) must be selected.

P45/UO:

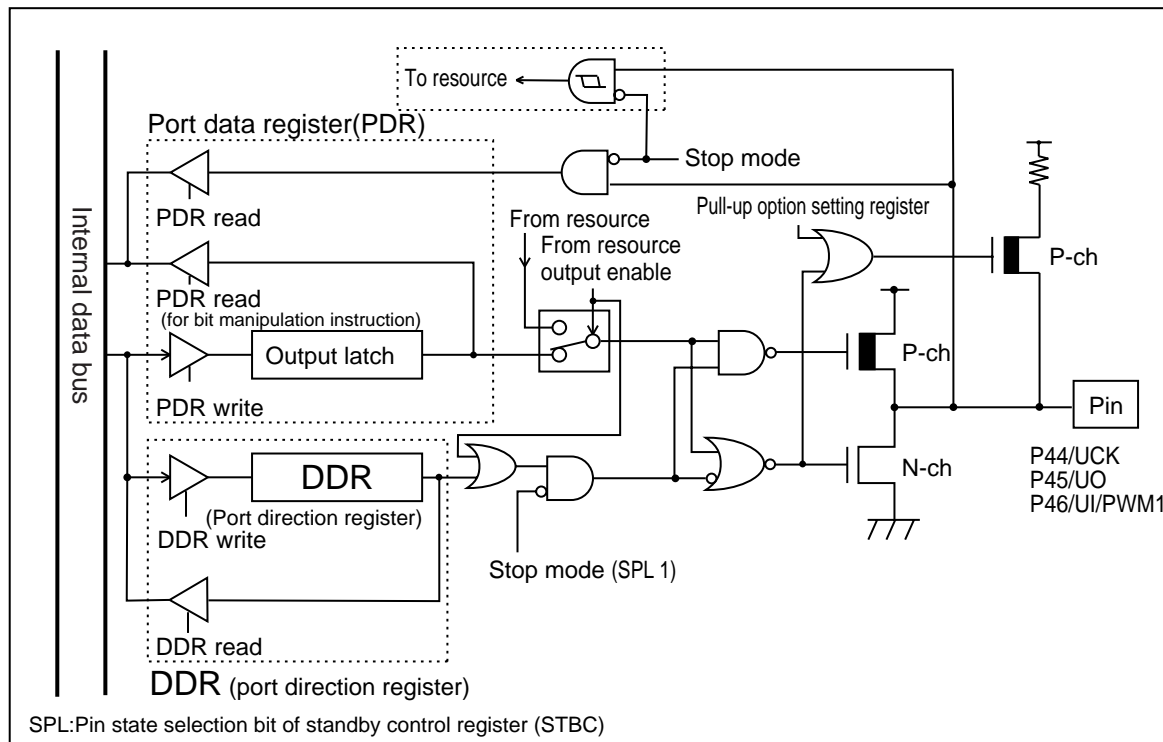
Functions as a general-purpose input and output port (P45) and UART/SIO serial output pin (UO). When serial data output is enabled, the pin functions as the UART/SIO serial data output pin (UO) regardless of the value of the corresponding port direction register.

P46/UI/PW1:

Functions as a general-purpose input and output port (P46) and UART/SIO data input pin (hysteresis input) (UI). When this pin is used as the UART/SIO serial data input pin, set it as the input port (DDR4: bit 6 = 0) using the corresponding port direction register.

■ Block Diagram of the Pins Related to UART/SIO

Figure 11.3-1 Block Diagram of the Pins Related to UART/SIO



When pull-up resistance is selected for the pull-up option setting, the pin state in stop mode (SPL = 1) will be the H level (pull-up state) instead of high impedance. However, pull-up will be invalid during reset, and will be Hi-Z.

11.4 Registers of the UART/SIO

This section explains the registers related to UART/SIO.

■ UART/SIO Register (SCS, SMC1, SMC2, SSD, SIDR, SODR, SRC)

Figure 11.4-1 Registers related to UART/SIO

SCS (Serial clock switch register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 D _H	—	—	—	—	—	—	—	EXBRE	XXXXXXX _B
								R/W	
SMC1 (Serial mode control register 1)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 2 F _H	MD	PEN	TDP	SBL	CL	CS2	CS1	CS0	0000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
SMC2 (Serial mode control register 2)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 0 _H	RERC	RXE	TXE	BRGE	TXOE	SCKE	RIE	TIE	0000000 _B
	R/W	R/W	R/W	W	R/W	R/W	R/W	R/W	
SSD (Serial status and data register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 1 _H	PER	OVE	FER	RDRF	TDRE	-	-	-	00001XXX _B
	R	R	R	R	R				
SIDR (Serial input data register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 _H									XXXXXXXX _B
	R	R	R	R	R	R	R	R	
SODR (Serial output data register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 _H									XXXXXXXX _B
	W	W	W	W	W	W	W	W	
SRC (Serial rate control register)									
Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 3 _H									XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and writable
 R : Read only
 W : Write only
 — : Unused
 X : Undefined

11.4.1 Serial Mode Control Register 1 (SMC1)

The serial mode control register 1 (SMC1) controls the UART/SIO operating mode. The register sets whether parity is used, number of stop bits, operating mode (data length), synchronous or asynchronous mode, and the serial clock.

■ Serial Mode Control Register 1 (SMC1)

Figure 11.4-2 Serial Mode Register 1 (SMC1)

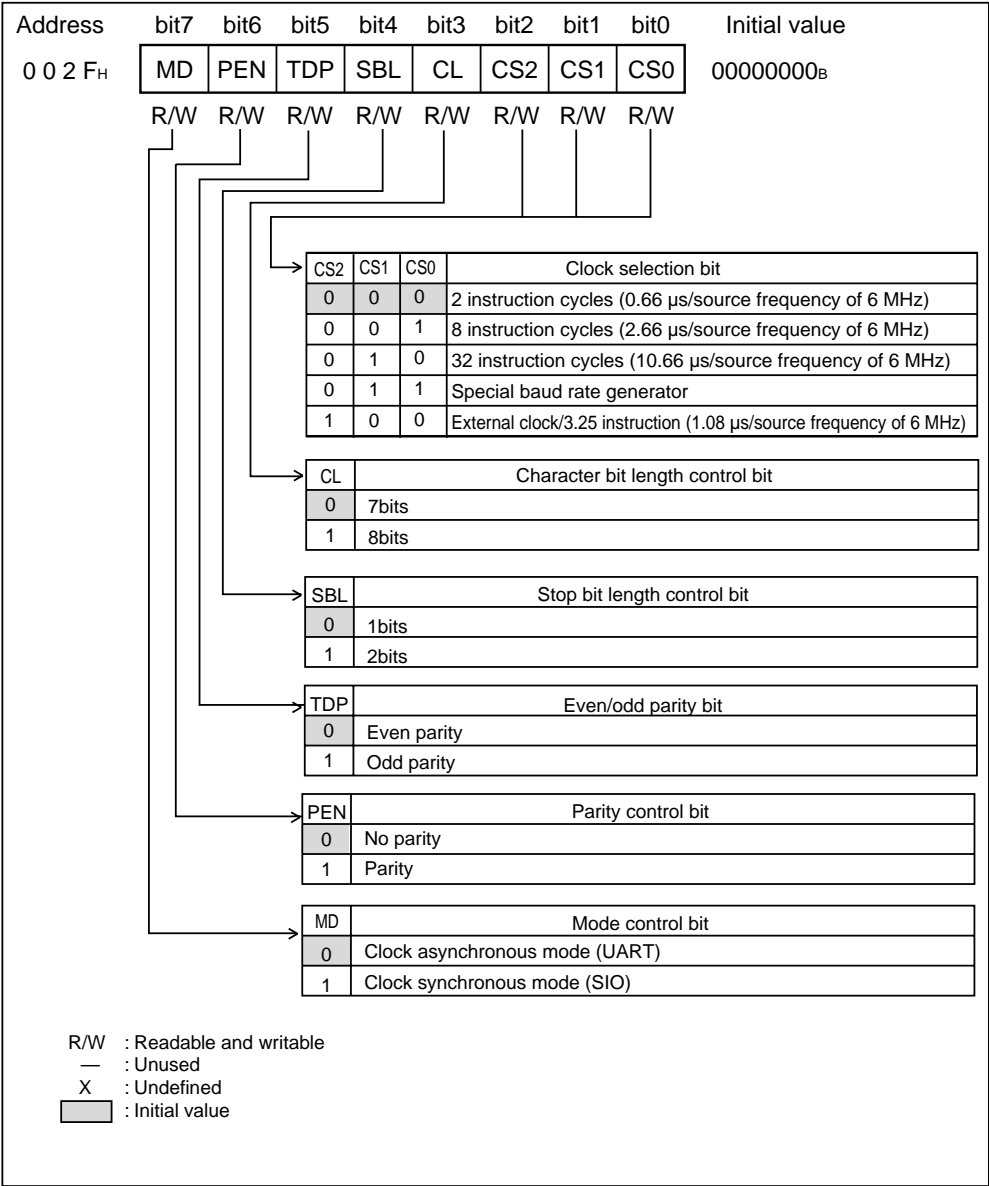


Table 11.4-1 Functions the Serial Mode Control Register 1 (SMC1) Bits

Bit name		Function
bit7	MD: Mode control bit	Specifies the UART operating mode. In asynchronous mode, the divide-by-eight clock of the serial clock is used. In clock synchronous mode, the selected serial clock is used.
bit6	PEN: Parity control bit	Specifies whether the parity bit is used in the clock asynchronous mode.
bit5	TDP: Even/odd parity bit	Specifies the parity data to be inserted when the serial interface signal is sent in the clock asynchronous mode. The parity data check is performed when the serial interface signal is received.
bit4	SBL: Stop bit length control bit	Specifies the stop bit length to be used in the clock asynchronous mode. The stop bit with the specified length is inserted when the serial interface signal is sent. The stop bit is determined using one bit length regardless of the set value when the serial interface signal is received.
bit3	CL: Character bit length control bit	Specifies the number of character bits to be used in the clock asynchronous mode.
bit2 bit1 bit0	CS1, CS2, CS0: Clock selection bit	Selects the serial clock. Selection of either the external clock or the internal clock when CS2, CS1, CS0 = 100 _B is set in SCS: EXBRE.

11.4.2 Serial Mode Control Register 2 (SMC2)

The serial mode control register 2 (SMC2) controls the UART/SIO operating mode. The registers sets serial clock output enable or disable, serial data output enable or disable, serial port and general-purpose port switching, and enabling or disabling of interrupts.

■ Serial Mode Control Register 2 (SMC2)

Figure 11.4-3 Serial Mode Register 2 (SMC2)

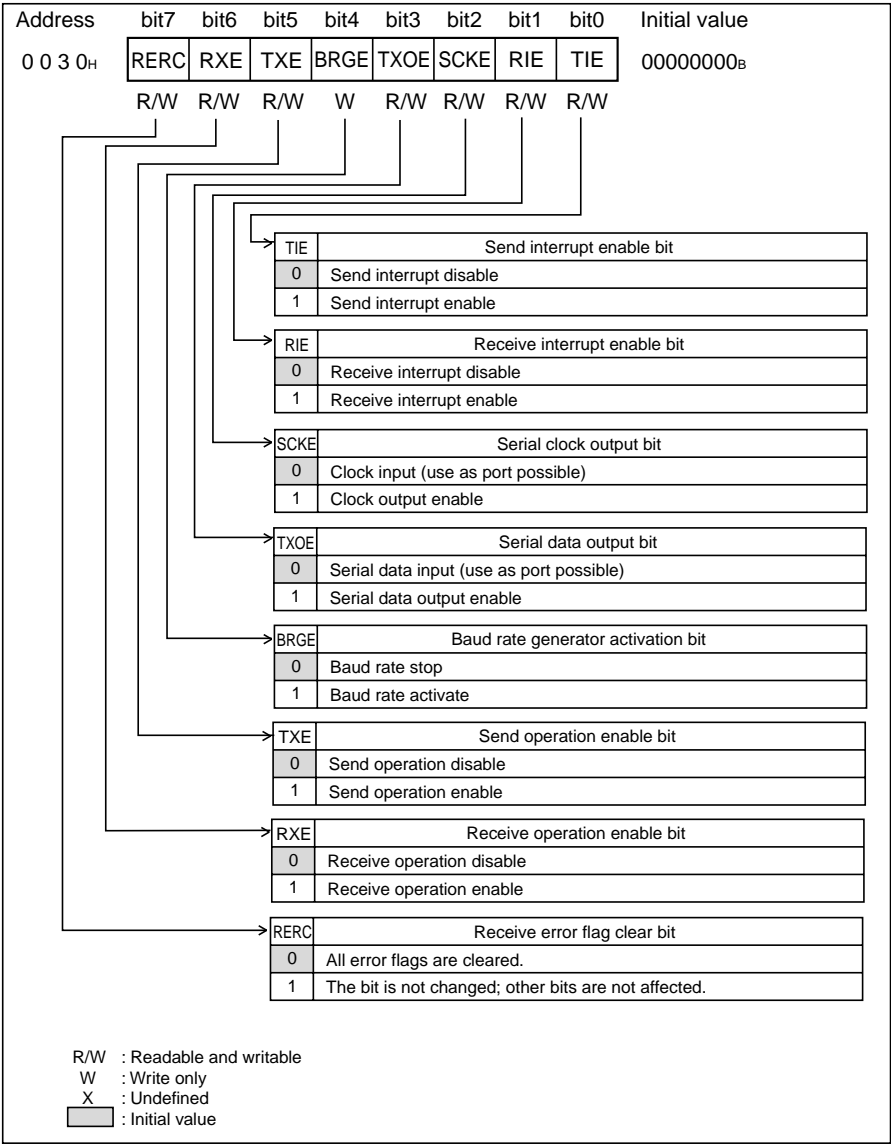


Table 11.4-2 Functions of the Serial Mode Control Register 2 (SMC2) Bits

Bit name		Function
bit7	RERC: Receive error clear bit	All SSD register error flags (PER/OVR/FER) are cleared when 0 is written to this bit. 0 is always read.
bit6	RXE: Receive operation enable bit	This bit enables serial data receive. When 0 is written to this bit during a receive operation, operation stops after the data is received, and the receive operation is disabled.
bit5	TXE: Send operation enable bit	This bit enables serial data send. When 0 is written to this bit during a send operation, operation stops after the data is sent, and the send operation is disabled.
bit4	BRGE: Baud rate generator activation bit	This bit activates the baud rate generator.
bit3	TXOE: Serial data output bit	This bit controls the enabling and disabling of serial data output.
bit2	SCKE: Serial clock output bit	This bit controls the input and output of the serial clock in clock synchronous mode. When inputting the external clock to P44/UCK, set the pin to input (bit 4 = 0 of DDR4).
bit1	RIE: Receive interrupt enable bit	This bit enables receive interrupts. A receive interrupt is immediately issued when receive interrupts are enabled and the RDRF bit is 1 or all error flags are 1.
bit0	TIE: Send interrupt enable bit	This bit enables send interrupts. A send interrupt is immediately issued when send interrupts are enabled and the TDRE bit is 1.

11.4.3 Serial Clock Switch Register (SCS)

The serial clock switch register (SCS) controls the switching of the input clock of the special baud rate generator and clock input to the selector circuit.

■ Serial Clock Switch register (SCS)

Figure 11.4-4 Serial Clock Switch Register (SCS)

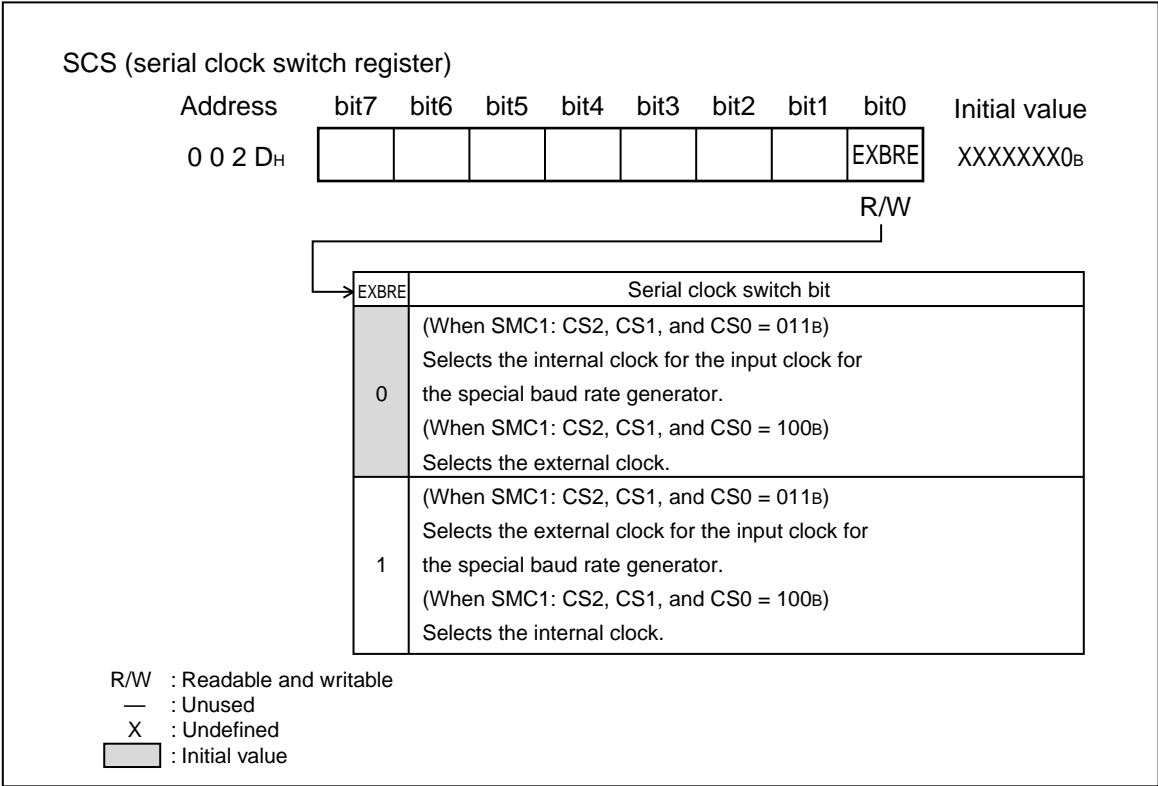


Table 11.4-3 Functions of the Serial Clock Switch Register (SCS) Bits

Bit name		Function
bit7	EXBRE: Clock switch bit	Sets the switching of the input clock of the special baud rate generator. Sets switching between the external clock and the internal clock (3.25 instruction cycles) for clock input when SMC1: CS2, CS1, CS0 = 100B is selected.

11.4.4 Serial Status And Data Register (SSD)

The serial status and data register (SSD) indicates the status of UART/SIO receive and send, and the error status.

■ Serial Status and Data Register (SSD)

Figure 11.4-5 Serial Status And Data Register (SSD)

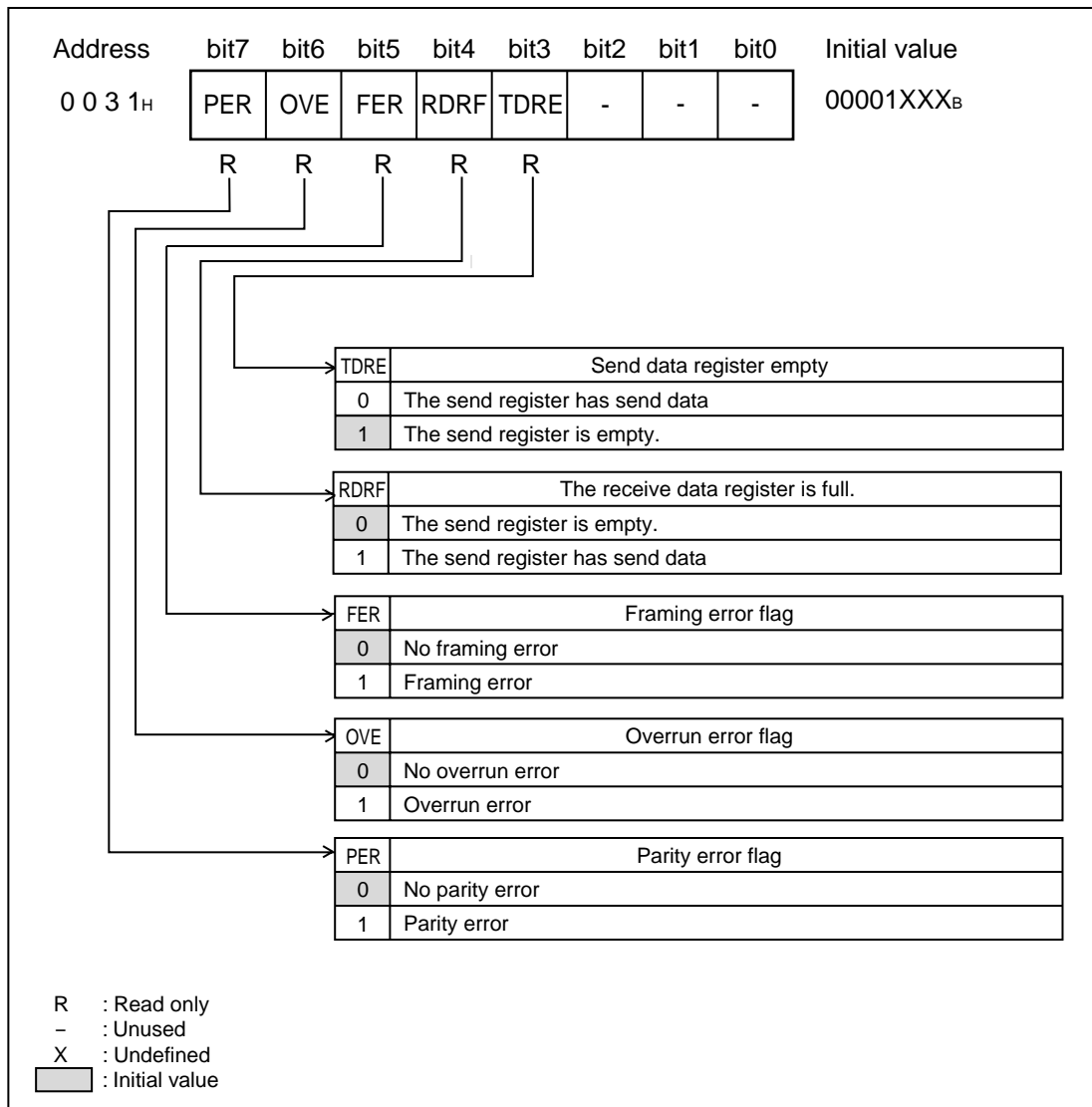


Table 11.4-4 Functions of the Serial Status and Data Register (SSD) Bits

Bit name		Function
bit7	PER: Parity error flag	The parity error flag is set when a parity error is issued at the receive destination, and is cleared by writing 0 to the RERC bit of the SMC2 register. SIO data is invalid when this flag is set. An interrupt is issued if the PER bit is set when the RIE bit is 1.
bit6	OVE: Overrun error flag	The overrun error flag is set when an overrun error is issued during receive, and by writing 0 to the RERC bit of the SMC2 register. SIO data is invalid when this flag is set. An interrupt is issued if the OVE bit is set when the RIE bit is 1.
bit5	FER: Framing error flag	The framing error flag is set when a framing error is issued at receive, and by writing 0 to the RERC bit of the SMC2 register. SIO data is invalid when this flag is set. An interrupt is issued if the FER bit is set when the RIE bit is 1.
bit4	RDRF: Receive data register full	This flag indicates the status of the receive data register (SIDR). The flag is set when receive data is loaded into the SIDR register, and is cleared when the SIDR register is read. An interrupt is issued if the RDRF bit is set when the RIE bit is 1.
bit3	TDRE: Send register empty	This flag indicates the status of SODR (serial send data register). The flag is cleared when send data is written to the SODR register, and is set when that data is loaded into the send shifter and sending starts. An interrupt is issued if the TDRE bit is set when the TIE bit is 1.

11.4.5 Serial Input Data Register (SIDR)

The serial input data register (SIDR) is used for serial data input (receive).

■ Serial Input Data Register (SIDR)

Figure 11.4-6 "Serial Input Data Register (SIDR)" shows the bit configuration of the serial input data register.

Figure 11.4-6 Serial Input Data Register (SIDR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 _H									XXXXXXXX _B
	R	R	R	R	R	R	R	R	

R : Read only
X : Undefined

This register stores the received data. A serial data signal sent to the serial data input pin (UI pin) is converted in the shift register, and is stored in SIDR.

When receive data is correctly set in the register, the receive data flag bit (RDRF) is set to 1, and an interrupt is issued when receive interrupt requests have been enabled. When receive data is stored in the register after checking the RDRF bit during interrupt processing or in the program, the RDRF flag is cleared by reading the SIDR register.

11.4.6 Serial Output Data Register (SODR)

The serial output data register (SODR) is used for serial data output (send).

■ Serial Output Data Register (SODR)

Figure 11.4-7 "Serial Output Data Register (SODR)" shows the bit configuration of the serial output data register.

Figure 11.4-7 Serial Output Data Register (SODR)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 2 _H									XXXXXXXX _B
	W	W	W	W	W	W	W	W	

W : Write only
X : Undefined

When send data is written to this register after the SSD register is read in the send enable state, send data is transferred to the send shift register, converted to serial data, and sent from the serial data output pin (UO pin).

When send data is written to the SODR register, the send data flag bit is set to 0. When send data is transferred to the send shift register, the send data flag bit is set to 1, and the next send data can be written. Perform the next send data write when an interrupt is issued or when the send data flag bit is 1.

11.4.7 Serial Rate Control Register (SRC)

The serial rate control register (SRC) controls the UART/SIO data transfer speed (baud rate).

■ Serial Rate Control Register (SRC)

Figure 11.4-8 Serial Rate Register (SRC)

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
0 0 3 _H									XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable and writable
 X : Undefined

When the clock selection bits (CS2, CS1, CS0) are 011, the special baud rate generator is selected as the serial clock. This register sets the optional baud rate clock. Also, this register must be written to while the UART is stopped.

11.5 Interrupts of the UART/SIO

UART/SIO has three error flag bits related to interrupts (PER, OVE, FER), a receive data flag bit (RDRF), and a send data flag bit, and the following two interrupt causes:

- Transfer of receive data from the receive shift register to the serial input data register (SIDR)
- Transfer of send data from the serial output data register (SODR) to the send shift register

■ Send Interrupts

When output data is written to the SODR register after the SSD register is read, the data written to the SODR register is transferred to the internal send shift register. When next write is possible, the TDRE bit is set to 1, and if send interrupts are enabled (SMC2: TIE = 1), an interrupt request (IRQ6) is issued to the CPU.

■ Receive Interrupts

The RDRF bit is set to 1 when data is input correctly up to the stop bit. The error flag bits are set to 1 if an overrun parity framing error is issued.

These bits are set when a stop bit is detected, and if receive interrupts are enabled (SSD: RIE = 1), an interrupt request (IRQ6) is issued to the CPU.

■ Register and Address of Vector Table Related to UART/SIO Interrupts

Table 11.5-1 Register and Address of Vector Table Related to UART/SIO Interrupts

Interrupt name	Interrupt level setting register			Address of vector table	
	Register	Setting bit		Higher	Lower
IRQ6	ILR2(007D _H)	L61(bit5)	L60(bit4)	FFEE _H	FFEF _H

See Section 3.4.2 "Processing During Interrupt Operation", for details of the interrupt operation.

11.6 Operation of the UART/SIO

This section explains the operation of UART/SIO.

UART/SIO has two normal mode serial communication functions (operating modes 0 and 1).

■ UART/SIO Operation

○ Operating modes

UART/SIO has two operating modes, clock synchronous (SIO) and asynchronous (UART). (See Table 11.1-1 "UART/SIO Operating Modes".)

11.7 Explanation of Operating Mode 0

Operating mode 0 operates as clock asynchronous mode (UART).

■ Explanation of UART/SIO Operating Mode 0

Selection of the serial clock is done with CS2 to CS0 of the SMC1 register. The serial clock is selected from five output types: three internal clocks, an external clock, and the baud rate generator. The clock must be input continuously when external clock is selected.

In CLK asynchronous mode, the shift clock selected in CS2 to CS0 is the divide-by-eight clock, and transfer is possible from - 2% to + 2% of the selected baud rate. Baud rate calculations by the internal and external clock and baud rate generator are described below.

○ Baud Rate Calculation by Internal and External Clock

When SMC1 (CS2, CS1 and CS0 = 000_B, 001_B, 010_B) is selected and the EXBRE bit of SCS (serial switching generator) is set to 0 with CS2, CS1, and CS0 = 100_B, baud rates can be calculated using the following equation:

$$\text{Baud rate value} = \frac{1}{8 \times \boxed{\text{Clock cycle selected with CS2 to CS0}}} \quad [\text{bps}]$$

○ **Baud Rate Calculation Equation when SMC1 (CS2, CS1, and CS0 = 100_B) is Selected and the EXBRE Bit is Set to 1**

When the EXBRE bit is set to 1 with CS2, CS1, and CS0 = 100_B, baud rates can be calculated using the following equation:

$$\text{Baud rate value} = \frac{1}{8 \times 13 \times 1 / F_{\text{CH}}} \quad [\text{bps}]$$

F_{CH} : Main clock oscillation frequency

Example: (F_{CH} : When Fch is 12 MHz)

$$\text{Baud rate value} = \frac{1}{8 \times 13 \times 1 / 12 \text{ MHz}} \approx 115384 \quad [\text{bps}]$$

○ **Baud Rate Calculation when Special baud Rate Generator is Used**

When the EXBRE bit is set to 0 with CS2, CS1, and CS0 = 011_B, baud rates can be calculated using the following equation:

$$\text{Baud rate value} = \frac{1}{8 \times 2 \times \begin{matrix} 64/F_{\text{CH}} \\ 16/F_{\text{CH}} \\ 8/F_{\text{CH}} \\ 4/F_{\text{CH}} \end{matrix} \times \begin{matrix} \text{SRC register} \\ \text{value (SRC)} \end{matrix}} \quad [\text{bps}]$$

Clock gear selection

F_{CH} : Main clock oscillation

○ External Clock Selection when Special Baud Rate Generator is Used

When the EXBRE bit is set to 1 with CS2, CS1, and CS0 = 011B, baud rates can be calculated using the following equation:

Baud rate value =

1

$8 \times 4 \times 2 \times 1/F_{ex}$

SRC register value (SRC)

[bps]

Fex : External clock frequency

Table 11.7-1 Example: Asynchronous transfer rate determined by baud rate generator

Frequency used ^{*1}	12MHz	10MHz	8MHz	7.3728MHz	4.9152MHz
Baud rate The table in parentheses is the SRC register setting value. ^{*2}	-	78125(n=2)	-	-	76800(n=1)
	-	39062(n=4)	-	38400(n=3)	38400(n=2)
	-	19531(n=8)	-	19200(n=6)	19200(n=4)
	-	9765(n=16)	9615(n=13)	9600(n=12)	9600(n=8)
	4807(n=39)	4882(n=32)	4807(n=26)	4800(n=24)	4800(n=16)
	2403(n=78)	2403(n=65)	2403(n=52)	2400(n=48)	2400(n=32)
	1201(n=156)	1201(n=130)	1201(n=104)	1200(n=96)	1200(n=64)
	-	-	600(n=208)	600(n=192)	600(n=128)

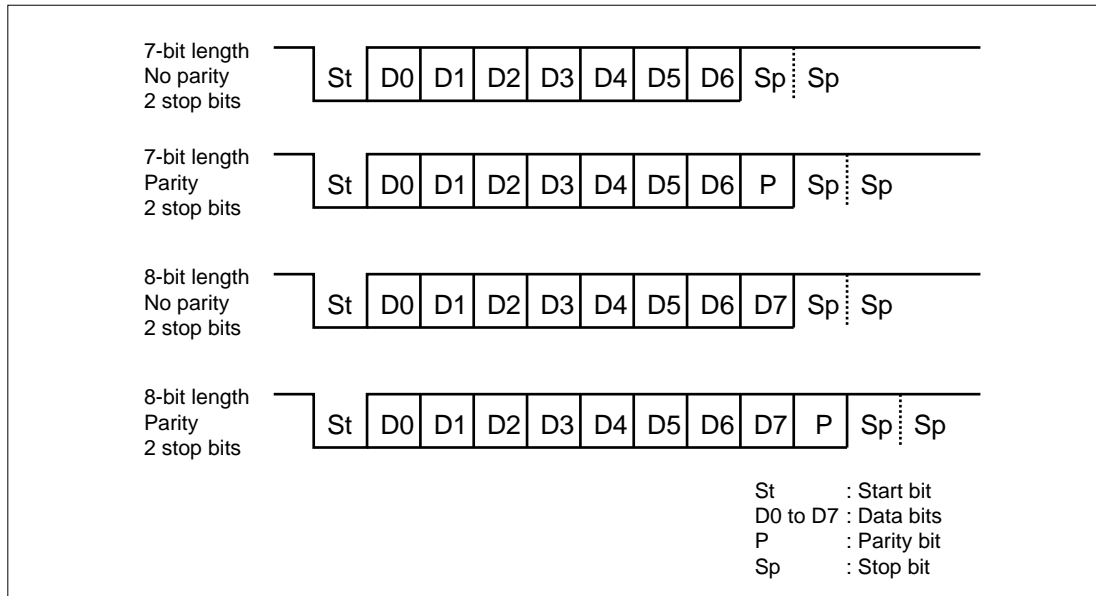
^{*1}: Clock frequency when special baud rate generator is used (Fch,Fex)
^{*2}: Example of clock gear maximum speed when the internal clock is selected

■ Transfer Data Format

The UART only handles data with in NRZ (Non Return to Zero) format. The following diagram shows the data format when the stop bit length is 2 bits.

Data transfer always starts with the start bit (L level), and continues according to the data bit length in LSB first order, and is terminated with the stop bit (H level). Idle is H level.

Figure 11.7-1 Transfer Data Format

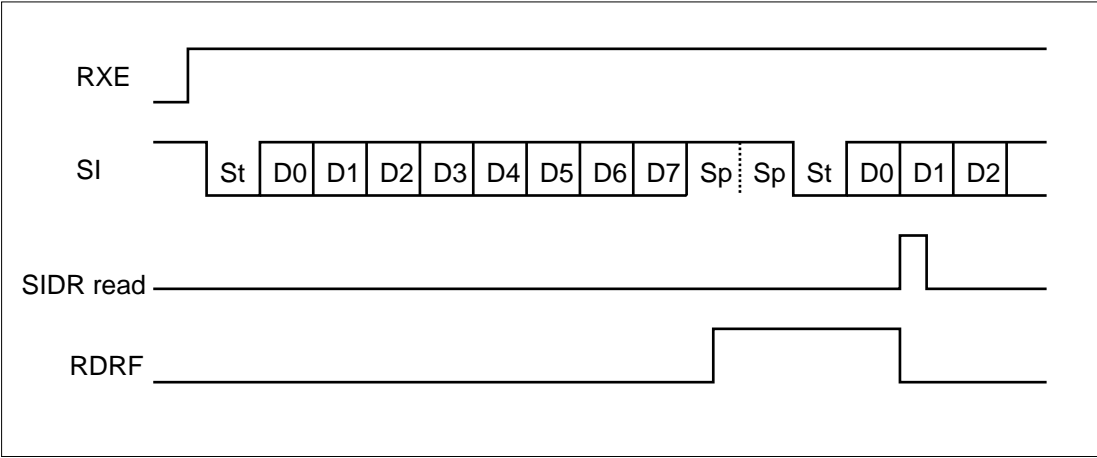


■ Receive Operation in CLK Asynchronous Mode

The CS2 to CS0 bits of the SMC1 register select the baud rate clock. (See clock selection for the baud rate clock.) Receive operation is enabled when the RXE bit is 1, and receive operation starts on the first falling edge of the input data (detection of start bit). When the receive operation is completed, the RDRF bit of the SSD register is set to 1, and receive data is loaded into the SDR register. Also, a receive interrupt is issued to the CPU when the RDRF bit is set to 1 and the RIE bit is 1. If one of three errors (PER, OVE, or FER) has occurred, the RDRF bit is not set to 1, receive data is not loaded into SDR, and the value of the SDR register is the data that was previously received. Also, unless the RXE bit is set to 0, the receive operation that detects the start bit continues.

When 0 is written to the RXE bit of the SMC2 register during the receive operation, receive operation is disabled after data receive is terminated.

Figure 11.7-2 Receive Operation in CLK Asynchronous Mode

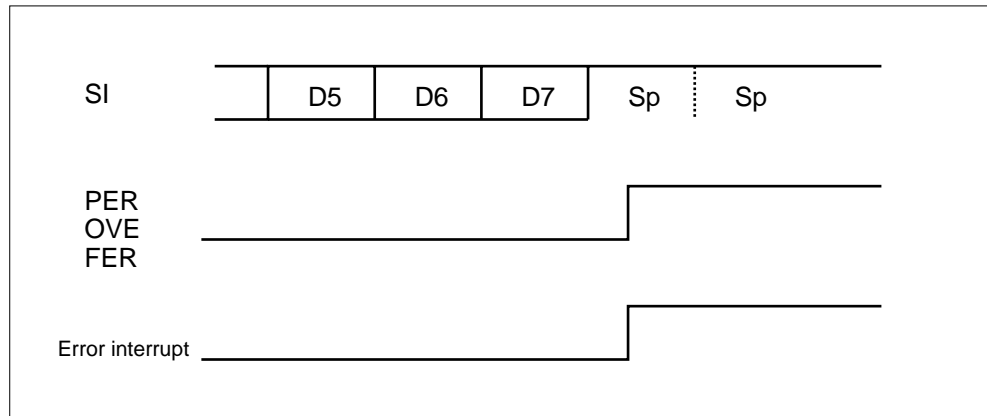


Receive Error in CLK Asynchronous Mode

Three types of errors are detected in CLK asynchronous mode: parity, overrun, and framing. When one of these errors is detected, the PER, OVE, or FER of the SSD register is set to 1.

When an error is detected, the value of the SDR register is the data that was previously received, since RDRF is not set and receive data is not loaded into the SDR register. Also, all error flags are cleared by writing 0 to the PERC bit of the SCM2 register.

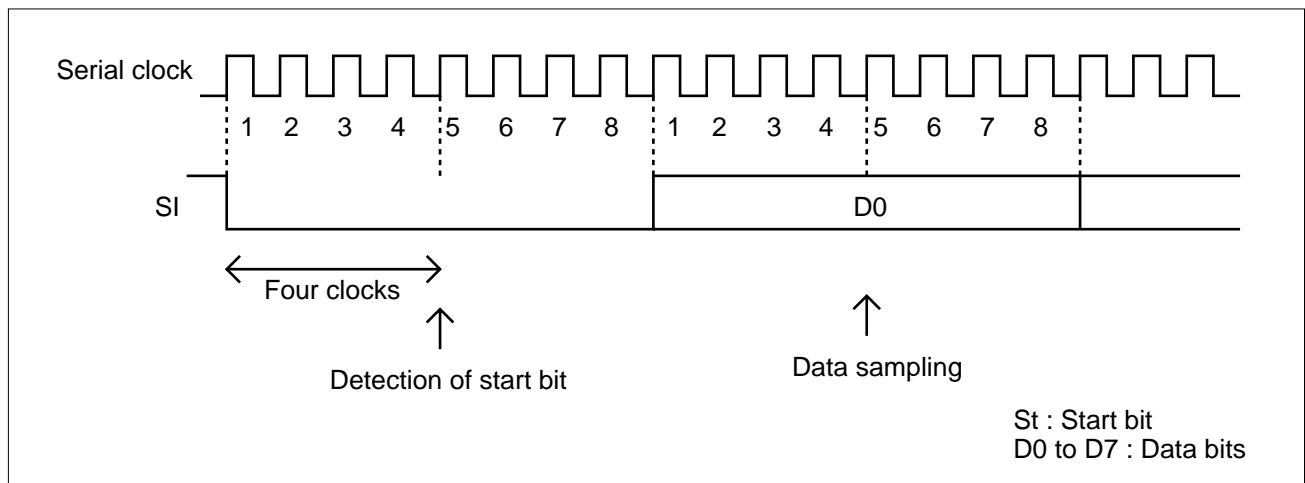
Figure 11.7-3 Timing for Setting a Receive Error



Start Bit Detection in a Receive Operation

The start bit is detected when the L level continues for four clocks for the selected serial clock (generator output, etc.) after the first falling edge of the input data. After detection of the start bit, data is sampled on the fifth rising edge of the serial clock.

Figure 11.7-4 Detection of Start Bit

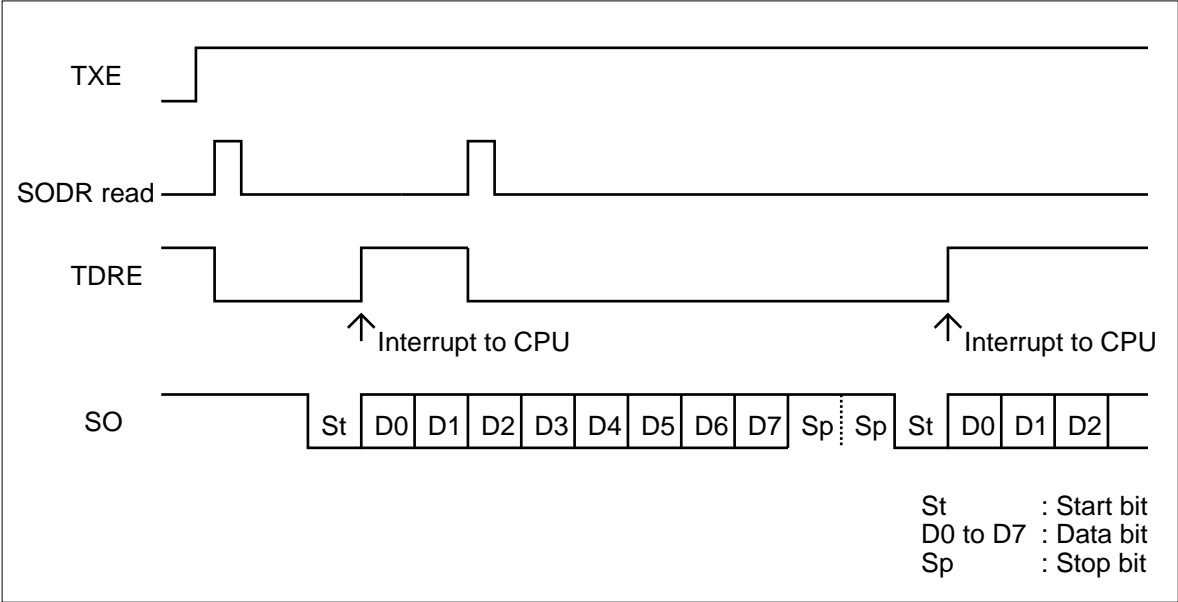


■ Send Operation in CLK Asynchronous Mode

When the TXE bit of the SMC2 register is 1 and send data is written to the SODR register, the TDRE bit of the SSD register is cleared and the send operation is started. When send data is output after SODR register data is loaded into the shifter, the TDRE bit of the SSD register is set. When data is written to the SODR register during the send operation (when the TDRE bit is 1), the TDRE bit is cleared and data is sent continuously after the specified bit length is sent.

Also, when 0 is written to the TXE bit of the SMC2 register during the send operation, the send operation is disabled after the specified bit length send operation when the SODR register empty (the TDRE bit is 1). When there is data (the TDRE bit is 1) in the SODR register, the send operation is disabled after the data in the SODR register is sent.

Figure 11.7-5 Send Operation in CLK Asynchronous Mode



11.8 Explanation of Operating Mode 1

Operating mode 1 operates in clock synchronous mode.

■ Explanation of UART/SIO Operating Mode

CLK synchronous mode operates according to the CS2 to CS0 bits of the SMC21 register, from which is selected one of three types of internal clock, the external clock, or baud rate generator output. Shifting operates using the selected clock as the shift clock. Set the SCKE clock to 0 when the external clock is input. Also, set the SCKE bit to 1 when outputting the internal clock or baud rate generator output. The following diagrams show the baud rate calculations for the internal clock, external clock, and baud rate generator.

○ Baud Rate Calculations for Internal and External Clocks

When SMC1 (CS2 = 000_B, CS1 = 001_B, CS0 = 010_B) is selected and the EXBRE bit of SCS (serial switching generator) is set to 0 with CS2, CS1, and CS0 = 100_B, baud rates can be calculated using the following equation:

$$\text{Baud rate value} = \frac{1}{\text{Clock cycle selected with CS2 to CS0}} \quad [\text{bps}]$$

○ Baud Rate Calculation Equation when SMC1 (CS2, CS1, and CS0 = 100_B) is Selected and the EXBRE Bit is Set to 1

When the EXBRE bit is set to 1 with CS2, CS1, and CS0 = 100_B, baud rates can be calculated using the following equation:

$$\text{Baud rate value} = \frac{1}{13 \times 1 / F_{\text{CH}}} \quad [\text{bps}]$$

F_{CH} : Main clock oscillation frequency

○ **Baud Rate Calculation when Special Baud Rate Generator is Used an Internal Clock is Selected**

When the EXBRE bit is set to 0 with CS2, CS1, and CS0 = 011_B, baud rates can be calculated using the following equation:

Baud rate value =

1

2 ×

64/F_{CH}

16/F_{CH}

8/F_{CH}

4/F_{CH}

 ×

SRC register value (SRC)

[bps]

Clock gear selection

F_{CH} : Main clock oscillation

○ **Baud Rate Calculation when Special Baud Rate Generator is Used and the External Clock is Selected**

When the EXBRE bit is set to 0 with CS2, CS1, and CS0 = 011_B, baud rates can be calculated using the following equation:

Baud rate value =

1

4 × 2 × 1/F_{ex} ×

SRC register value (SRC)

[bps]

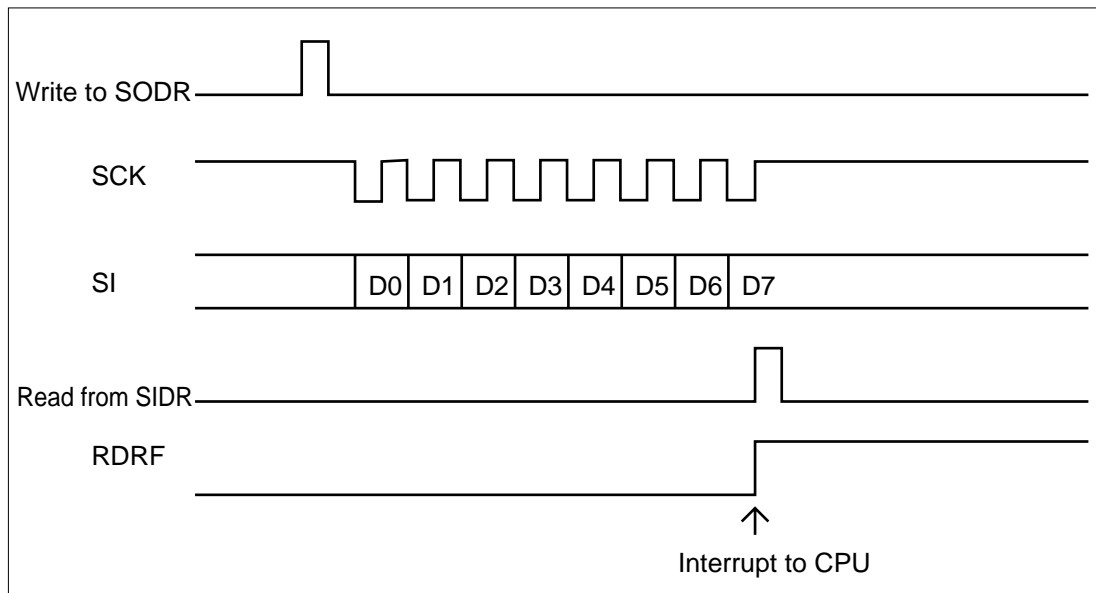
F_{ex} : External clock frequency

294

■ 8-bit Receive Operation

Receive operation is enabled by setting the TXE/RXE bits to 11. It starts by writing to the SODR register and is synchronized on the rising edge of the shift clock. When 8-bit data is received, shift data is loaded into the SIDR register, the RDRF flag is set to 1, and an interrupt request is issued to the CPU. If an overrun error is issued when the receive terminates, data is not loaded into the SIDR register. If 0 is written to the RXE bit during the receive operation, operation is stopped after the 8-bit data is received. Serial clock input is always H level in the serial operation stop state (regardless of the value of the RXE bit).

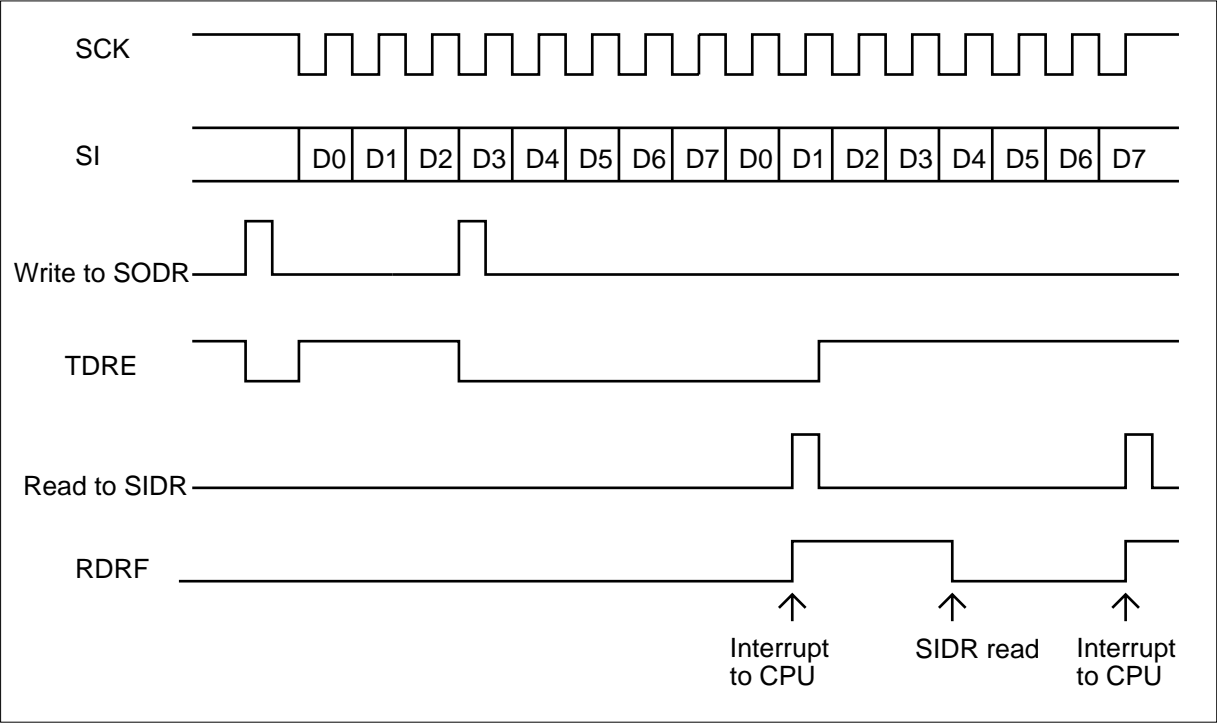
Figure 11.8-1 8-bit Receive Operation in CLK Synchronous Mode



■ Continuous Receive Operation

In addition to 8-bit data receive in CLK synchronous mode, continuous receive operation is also possible. In addition to the bit used for 8-bit receive, the TIE bit of the SMC2 register and the TDRE bit of the SSD register are used. The receive operation is enabled by setting the TXE/RXE bits to 11. Operation starts by writing to the SODR register and is synchronized on the rising edge of the shift clock. The TDRE bit is set to 1 when the shift operation starts, and an interrupt is issued to the CPU when TIE = 1. The next operation is enabled by writing to the SODR register before the 8-bit shift operation is terminated, and the receive operation continues after the 8-bit data is received. When the 8-bit data receive is completed, shift data is loaded into the SIDR register, the RDRF flag is set to 1, and an interrupt request is issued to the CPU when RIE = 1. If an overrun error is issued when the receive is completed, data is not loaded into the SIDR register, contains the previous receive data. Also, the receive interrupt (RDRF) is cleared when the SIDR register is read. The receive operation is stopped by writing 0 to the RXE bit. During a receive operation, the receive operation is stopped when 0 is written to the RXE bit after the 8-bit data is received.

Figure 11.8-2 Continuous Receive Operation in CLK Synchronous Mode

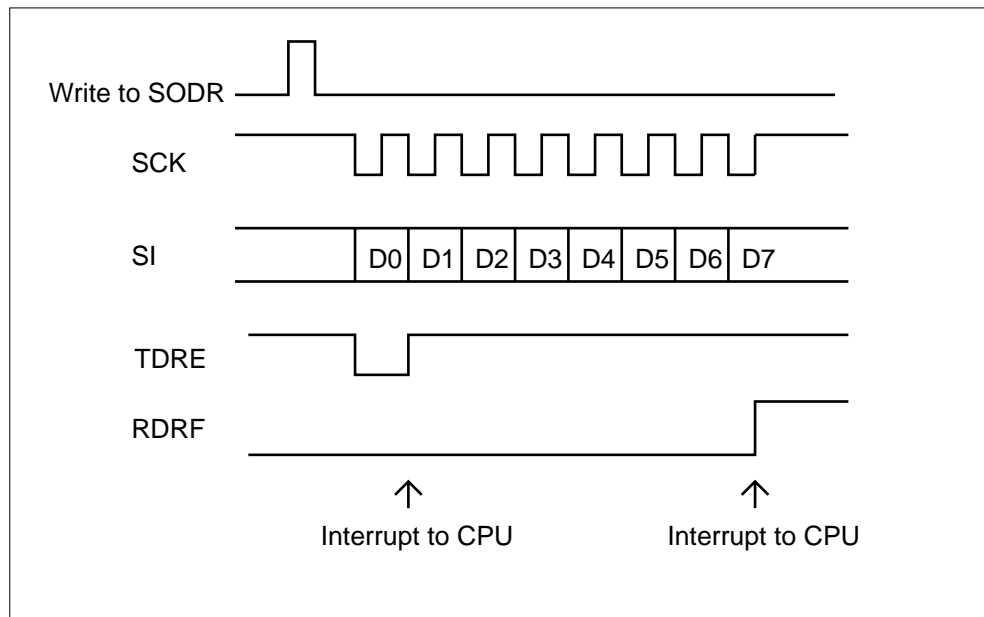


■ 8-bit Send Operation

The send operation starts by setting the TXE/RXE bits to 11 and writing to the SODR register. When the send operation starts, shifting is performed after the data written to the SODR register is loaded into the shifter. When the SODR register data is loaded into the shifter, the TDRE flag is set to 1, and an interrupt request is issued to the CPU when TIE = 1. Serial data is output by synchronization on the falling edge of the shift clock enabled by TXOE = 1.

The send operation stops after the 8-bit data is sent when 0 has been written to the TXE bit during the send operation. After an 8-bit send, the RDRF bit is set to 1, and an interrupt is issued to the CPU when RIE = 1. The transfer direction starts at bit 0 and terminates at bit 7. Serial clock input is always H level when serial operation is stopped (regardless of the value of the TXE bit).

Figure 11.8-3 8-bit Send Operation in CLK Synchronous Mode

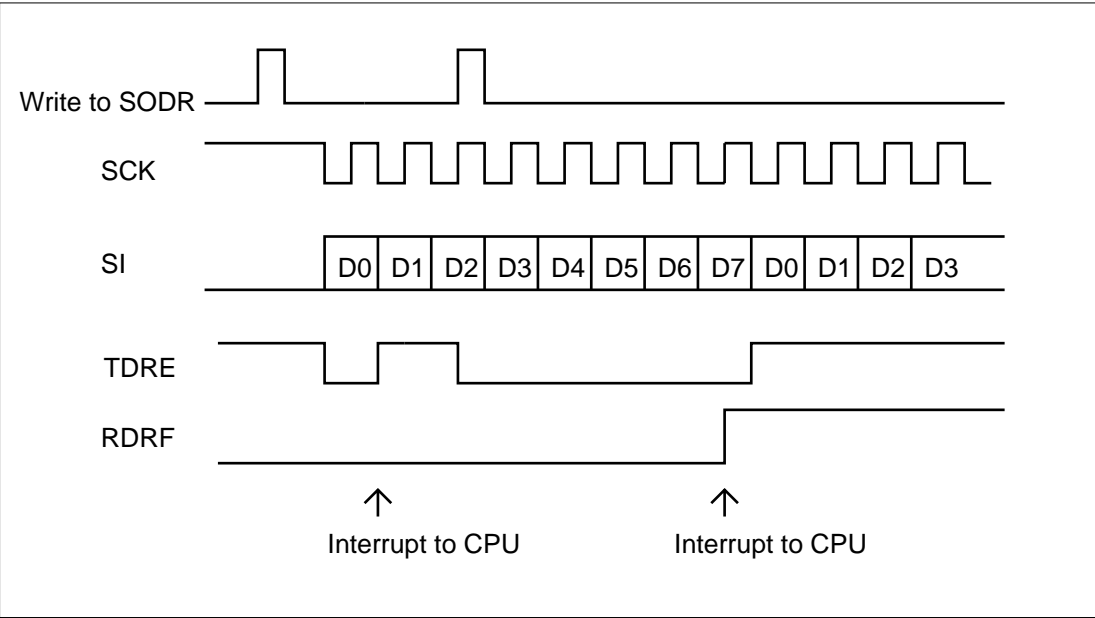


■ Continuous Send Operation

In addition to 8-bit data send in CLK synchronous mode, continuous send operation is also possible. This is done by setting the TXE/RXE bits to 1 and writing the data to the SODR register. When sending starts, shifting is performed after the data written to the SODR register is loaded into the shifter. When the SODR register data is loaded into the shifter, the TDRE flag is set to 1, and an interrupt request is issued to the CPU when TI = 1.

Continuous send is performed during the send operation by writing the next send data to the SODR register when the TDRE bit is 1 (the SODR register is empty). The send operation continues by writing to SODR to clear the TDRE bit, and loading the data written to the SODR register into the shifter after the 8-bit data is sent. Writing 0 to the TXE bit stops the sending of data. During a send operation, if 0 is written to the TXE bit, the send operation stops after the 8-bit data is sent when SODR register empty is indicated (the TDRE bit is 1). If there is data in the SODR register (the TDRE bit is 0), the send operation stops after the SODR register data is sent. The RDRF bit is set to 1 after 8-bit data is sent, and an interrupt is issued to the CPU when RIE = 1.

Figure 11.8-4 Continuous Send Operation in CLK Synchronous Mode



CHAPTER 12 CLOCK OUTPUT FUNCTION

This chapter explains the clock output function and the circuit operation.

12.1 "Overview of the Clock Output"

12.2 "Clock Output Pin (CLK)"

12.3 "Register of the Clock Output"

12.1 Overview of the Clock Output

The clock output function outputs from the port a waveform whose oscillation clock is multiplied and then divided by PLL.

■ Clock Output Function

The clock output function outputs from the port a waveform whose oscillation clock is multiplied and then divided by PLL. (The frequency is the same as the oscillation clock.)

The output port is shown below.

A waveform whose oscillation clock is multiplied and then divided: $P30/\overline{INT0}/CLK$

When setting the output of a waveform (CLK) whose oscillation clock is multiplied and then divided, set the external interrupt input enable bit to interrupt input disable (EIE: IE0 = 0).

12.2 Clock Output Pin (CLK)

This section explains the pins related to clock output (CLK) and contains a block diagram of the pins.

■ Pins Related to Clock Output (CLK)

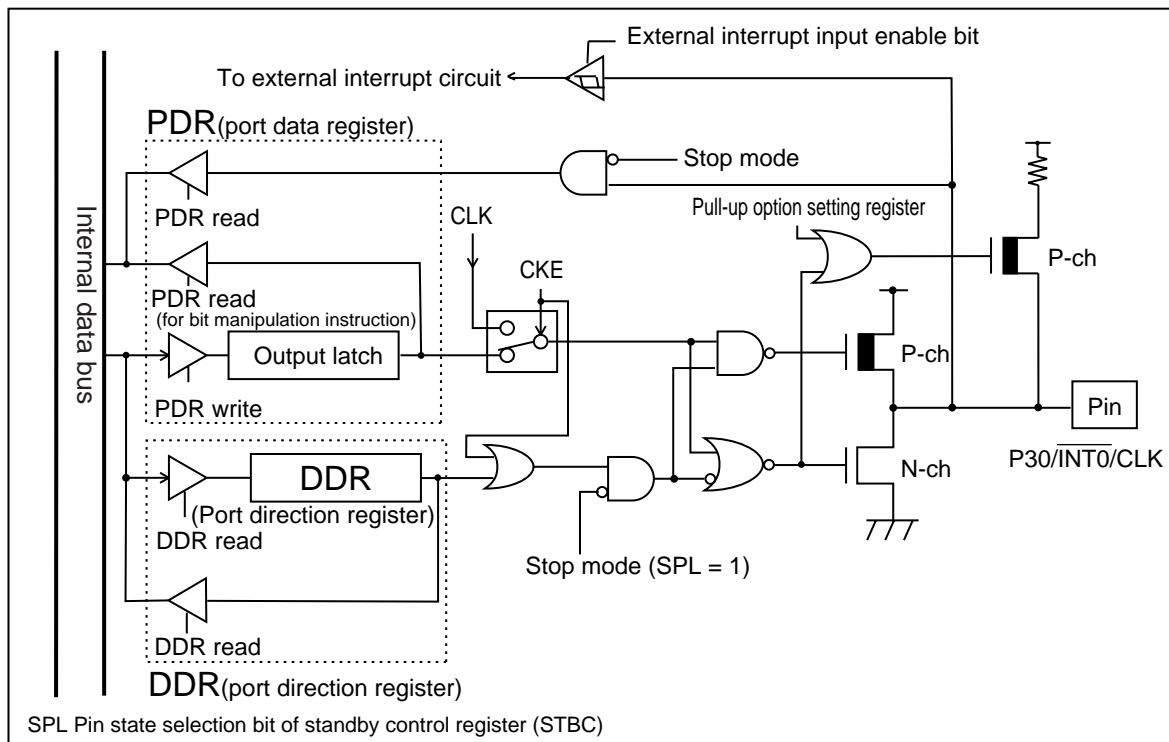
The pins related to clock output (CLK) are the P30/ $\overline{\text{INT0}}$ /CLK pins, which function as general-purpose input and output ports (P30), the external interrupt input pin ($\overline{\text{INT0}}$), and clock output port (CLK).

CLK:

The clock with a waveform whose frequency is the same as the source clock frequency (CLK) is output. When the clock (CLK) output with a frequency that is generated by doubling the waveform of the source clock frequency is specified, specify "disable" (EIE: IE0 = 0) for the external interrupt input enable bit.

■ Block Diagram of the Pins Related to Clock Output

Figure 12.2-1 Block Diagram of the Pins Related to Clock Output



Reference:

When pull-up resistance is selected for the pull-up option setting, the pin state in stop mode (SPL = 1) will be the H level (pull-up state) instead of high impedance. However, pull-up will be ineffective during reset, and the pin state will become Hi-Z.

12.3 Register of the Clock Output

This section explains the register related to clock output.

■ Register Related to Clock Output Control Register (CKR)

Figure 12.3-1 Register Related to Clock Output Control (CKR)

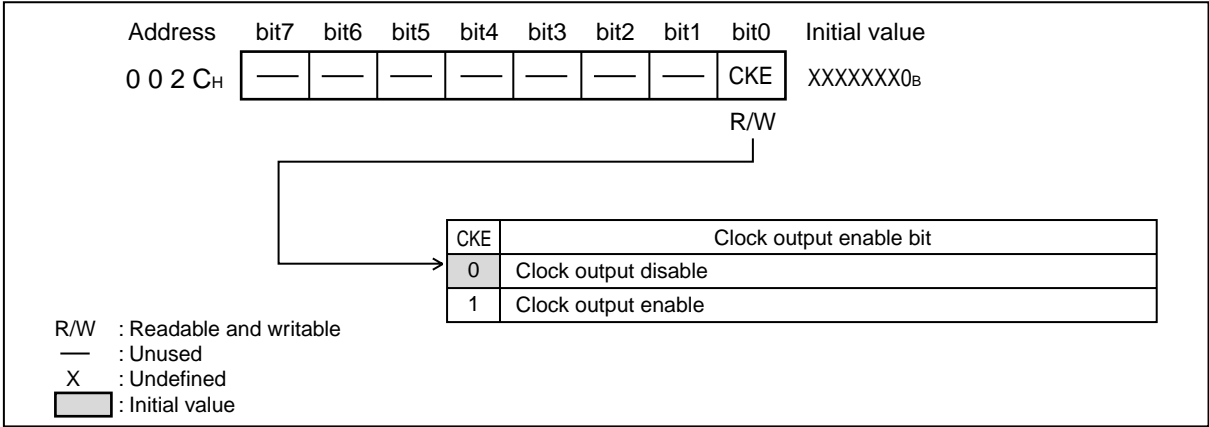


Table 12.3-1 Functions of the Clock Output Control Register (CKR) Bits

Bit name		Function
bit7 bit6 bit5 bit4 bit3 bit2 bit1	Unused bit	<ul style="list-style-type: none">Read values are undefinedAlways write 0.
bit0	CKE: Clock output enable bit	This bit enables output of the clock waveform.

CHAPTER 13 PULL-UP OPTION

This chapter explains the pull-up option.

13.1 "Overview of the Pull-up Option"

13.2 "Pull-up Option Setting Register"

13.1 Overview of the Pull-up Option

The pull-up option sets pull-up for ports 0, 1, 2, 3, 4, and 5.

■ Pull-up Option

The pull-up option causes the port 0, 1, 2, 3, 4, and 5 pins to be pulled up. Setting is done by setting (writing to) the pull-up option register.

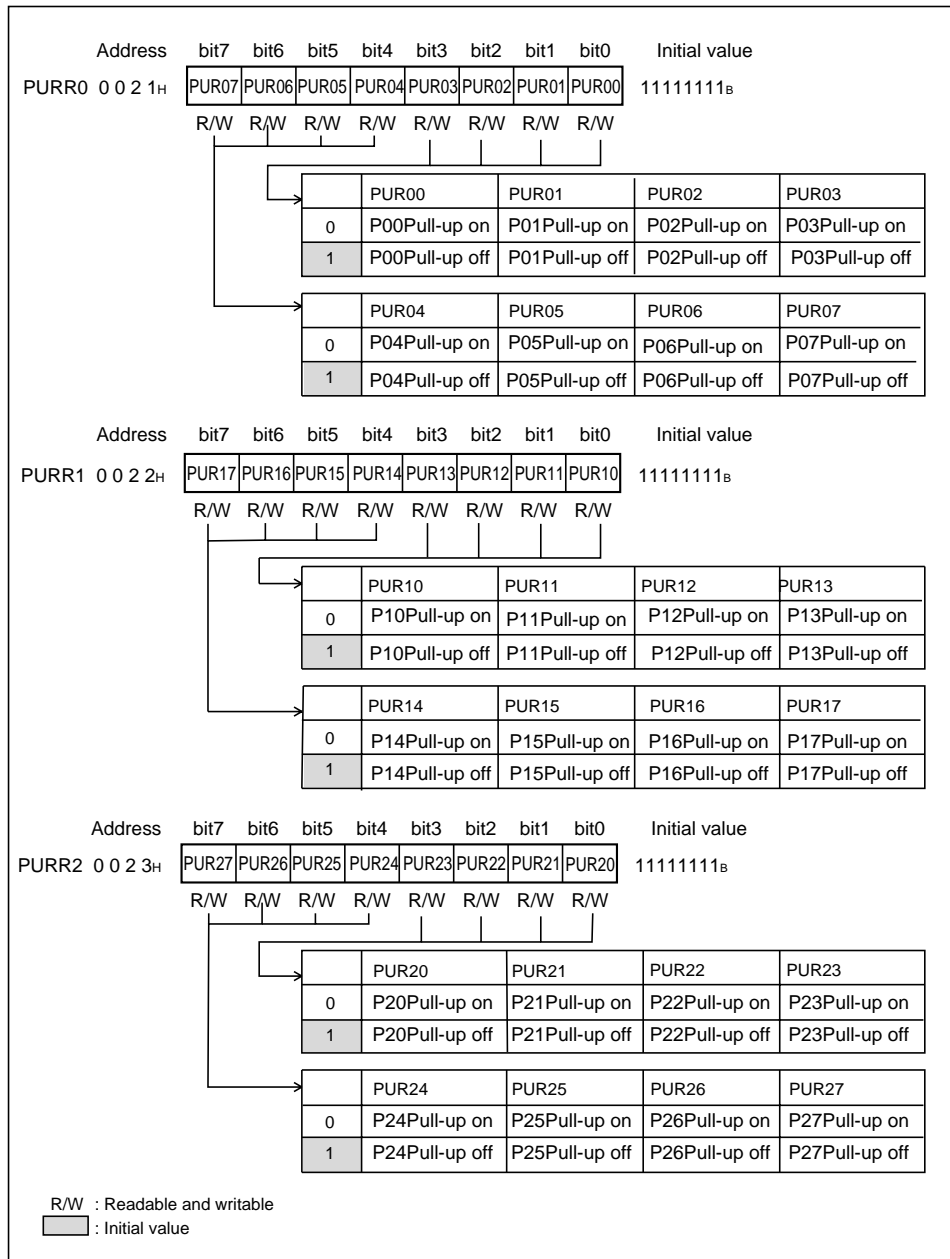
When pull-up resistance is selected for the pull-up option setting, the pin state in stop watch mode (SPL = 1) will be the H level (pull-up state) instead of high impedance. However, pull-up will be invalid during reset, and will be Hi-Z.

13.2 Pull-up Option Setting Register

This section explains the register that sets the pull-up option.

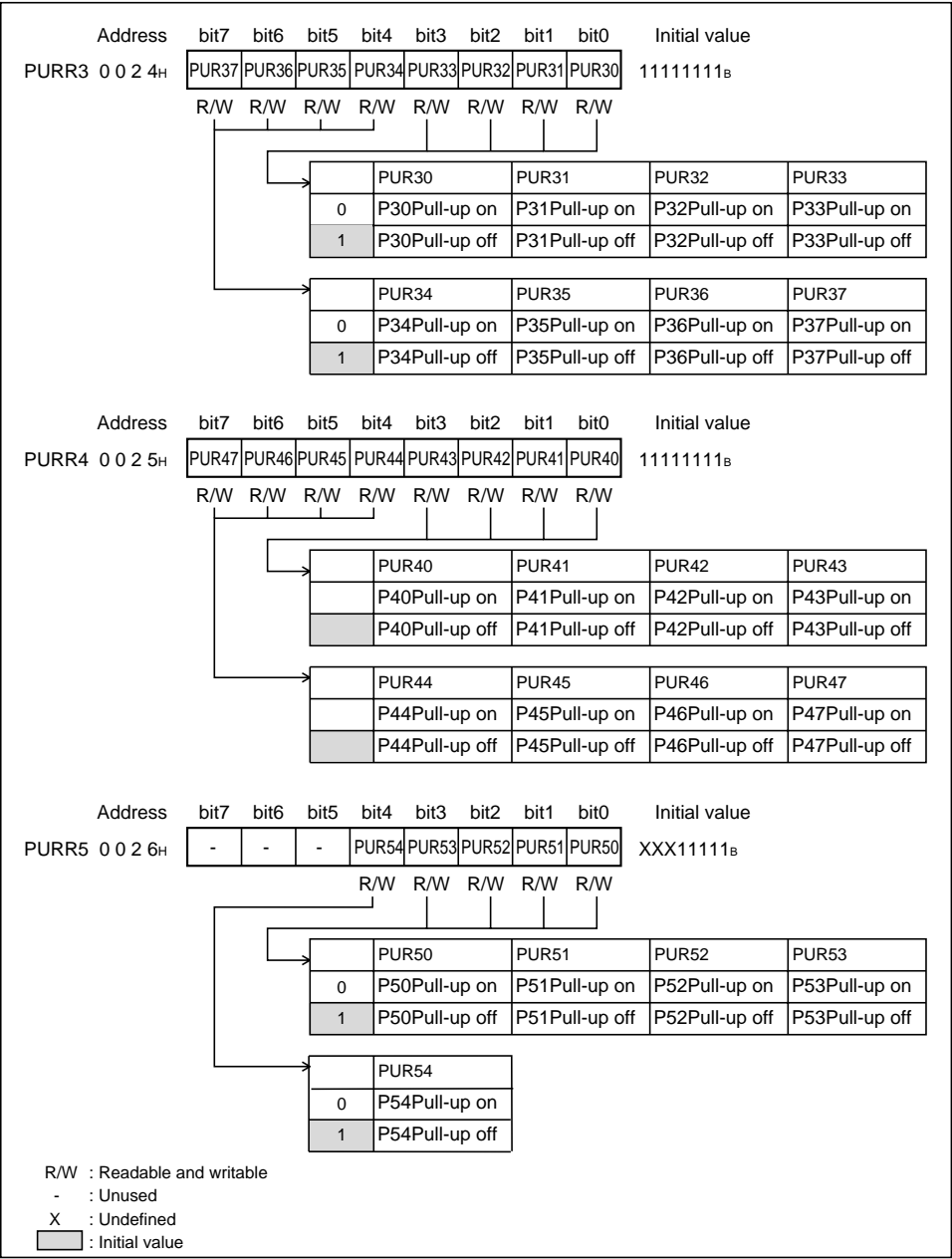
■ Pull-up Option Setting Registers (PURR 0, PURR 1, and PURR 2)

Figure 13.2-1 Pull-up Option Setting Registers (PURR 0, 1, and 2)



■ Pull-up Option Setting Registers (PURR 3, PURR 4, and PURR 5)

Figure 13.2-2 Pull-up Option Setting Registers (PURR 3, 4, and 5)



APPENDIX

The appendixes provide the I/O map, instruction lists, and other information.

APPENDIX A "I/O Map"

APPENDIX B "Instruction Overview"

APPENDIX C "Write Specifications for One-Time PROM and EPROM
Microcomputer"

APPENDIX D "Pin State of MB89580B/BW Series"

APPENDIX A I/O Map

The addresses described in Table A-1 are assigned to the peripheral registers provided in the MB89580.

■ I/O Map

Table A-1 I/O Map (1 / 3)

Address	Register abbreviation	Register name	Write, read	Initial value
00 _H	PDR0	Port 0 data register	R/W	XXXXXXXX
01 _H	DDR0	Port 0 data direction register	W	00000000
02 _H	PDR1	Port 1 data register	R/W	XXXXXXXX
03 _H	DDR1	Port 1 data direction register	W	00000000
04 _H	PDR2	Port 2 data register	R/W	XXXXXXXX
05 _H	Reserved			
06 _H	Reserved			
07 _H	SYCC	System clock control register	R/W	XXX11X00
08 _H	STBC	Standby control register	R/W	0001XXXX
09 _H	WDTC	Watchdog control register	R/W	0XXXXXXXX
0A _H	TBTC	Time base timer control register	R/W	00XXX000
0B _H	Empty area			
0C _H	PRR3	Port 3 data register	R/W	XXXXXXXX
0D _H	DDR3	Port 3 data direction register	R/W	00000000
0E _H	Reserved			
0F _H	Reserved			
10 _H	PDR4	Port 4 data register (general-purpose port operation)	R/W	XXXXXXXX
	PDR	Parallel port data register (parallel port operation)	R/W	XXXXXXXX
11 _H	DDR4	Port 4 data direction register (general-purpose port operation)	R/W	00000000
	PSR	Parallel port status register (parallel port operation)	R/W	XXXX0XXX
12 _H	PDR5	Port 5 data register	R/W	XXX111XX
13 _H	DDR5	Port 5 data direction register	R/W	XXXXXXXX00

Table A-1 I/O Map (2 / 3)

Address	Register abbreviation	Register name	Write, read	Initial value
14 _H	PDR6	Port 6 data register	R/W	XXXXXXXX
15 _H	PDCR	Parallel port data control register	R/W	XXX00000
16 to 20 _H	Reserved			
21 _H	PURR0	Port 0 pull-up option setting register	R/W	11111111
22 _H	PURR1	Port 1 pull-up option setting register	R/W	11111111
23 _H	PURR2	Port 2 pull-up option setting register	R/W	11111111
24 _H	PURR3	Port 3 pull-up option setting register	R/W	11111111
25 _H	PURR4	Port 4 pull-up option setting register	R/W	11111111
26 _H	PURR5	Port 5 pull-up option setting register	R/W	XXX11111
27 _H	CNTR1	PWM control register 1	R/W	00000000
28 _H	CNTR2	PWM control register 2	R/W	000X0000
29 _H	CNTR3	PWM control register 3	R/W	X000XXXX
2A _H	COMR1	PWM compare register 1	W	XXXXXXXX
2B _H	COMR2	PWM compare register 2	W	XXXXXXXX
2C _H	CKR	Clock output control register	R/W	XXXXXXXX0
2D _H	SCS	Serial clock switch register	R/W	XXXXXXXX0
2E _H	Reserved			
2F _H	SMC1	Serial mode control register 1	R/W	00000000
30 _H	SMC2	Serial mode control register 2	R/W	00000000
31 _H	SSD	Serial status and data register	R	00001XXX
32 _H	SIDR/SODR	Serial input/serial output data register	R/W	XXXXXXXX
33 _H	SRC	Serial rate control register	R/W	XXXXXXXX
34 _H to 3B _H	Reserved			
3C _H	EIE	External interrupt control register	R/W	00000000
3D _H	EIF	External interrupt flag register	R/W	XXXXXXXX0
3E _H , 3F _H	Reserved			
40 _H	PMDR	USB power mode register	R/W	XXXXXXXX0
41 _H to 4E _H	Reserved			
4F _H	DBARH	DMA base address register high	R/W	000000XX
50 _H	UMDR	USB reset mode register	R/W	1000XX00
51 _H	DBAR	DMA base address register	R/W	XXXXXXXX
52 _H	TDCR0	Transfer data count register 0	R/W	X0000000

APPENDIX A I/O Map

Table A-1 I/O Map (3 / 3)

Address	Register abbreviation	Register name	Write, read	Initial value
53 _H	TDCR11	Transfer data count register 11	R/W	00000000
54 _H	TDCR12	Transfer data count register 12	R/W	XXXXXX00
55 _H	TDCR21	Transfer data count register 21	R/W	00000000
56 _H	TDCR22	Transfer data count register 22	R/W	XXXXXX00
57 _H	TDCR3	Transfer data count register 3	R/W	X0000000
58 _H	UCTR	USB control register	R/W	00000000
59 _H	USTR1	USB status register 1	R/W	00000000
5A _H	USTR2	USB status register 2	R	XXXXXX00
5B _H	UMSKR	USB interrupt mask register	R/W	00000000
5C _H	UFRMR1	USB frame status register 1	R	XXXXXXXX
5D _H	UFRMR2	USB frame status register 2	R	XXXXXXXX
5E _H	EPER	USB end point enable register	R/W	XXXX0001
5F _H	EPBR0	End point 0 setup register	R/W	X0000000
60 _H	EPBR11	End point 1 setup register 11	R/W	0X000000
61 _H	EPBR12	End point 1 setup register 12	R/W	00000000
62 _H	EPBR21	End point 2 setup register 21	R/W	0X000000
63 _H	EPBR22	End point 2 setup register 22	R/W	00000000
64 _H	EPBR31	End point 3 setup register 31	R/W	XX0000XX
65 _H	EPBR32	End point 3 setup register 32	R/W	X0000000
66 _H to 7B _H	Reserved			
7C _H	ILR1	Interrupt level setting register 1	W	11111111
7D _H	ILR2	Interrupt level setting register 2	W	11111111
7E _H	ILR3	Interrupt level setting register 3	W	11111111
7F _H	Reserved			

○ Read/write

R/W: Readable and writable

R: Read only

W: Write only

○ Initial value

0: The initial value of this bit is "0".

1: The initial value of this bit is "1".

X: The initial value of this bit is undefined.

Note:

Do not use the registers of the Reserved areas.

APPENDIX B Overview of Instructions

Appendix B describes the instructions used by the F²MC-8L.

B.1 "Overview of F²MC-8L Instructions"

B.2 "Addressing"

B.3 "Special Instructions"

B.4 "Bit Manipulation Instructions (SETB, CLRB)"

B.5 "F²MC-8L Instructions"

B.6 "Instruction Map"

B.1 Overview of F²MC-8L Instructions

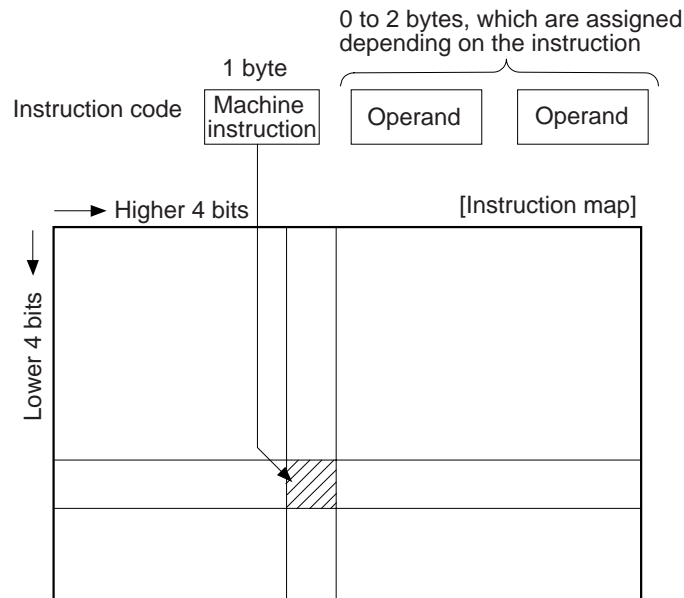
The F²MC-8L supports 140 types of instructions.

■ Overview of F²MC-8L Instructions

The F²MC-8L has 140 1-byte machine instructions (256-byte instruction map). An instruction code consists of an instruction and zero or more operands that follow.

Figure B.1-1 "Relationship between the Instruction Codes and the Instruction Map" shows the relationship between the instruction codes and the instruction map.

Figure B.1-1 Relationship between the Instruction Codes and the Instruction Map



- The instructions are classified into four types: transfer, arithmetic, branch, and other.
- A variety of addressing methods is available. One of ten addressing modes can be selected depending on the selected instruction and specified operand(s).
- Bit manipulation instructions are provided. They can be used for read-modify-write operations.
- Some instructions are used for special operations.

APPENDIX B Overview of Instructions

■ Symbols used with Instructions

Table B.1-1 "Symbols in the Instruction List" lists the symbols used in the instruction code descriptions in Appendix B.

Table B.1-1 Symbols in the Instruction List

Symbol	Meaning
dir	Direct address (8 bits)
off	Offset (8 bits)
ext	Extended address (16 bits)
#vct	Vector table number (3 bits)
#d8	Immediate data (8 bits)
#d16	Immediate data (16 bits)
dir:16	Bit direct address (8 bits:3 bits)
rel	Branch relative address (8 bits)
@	Register indirect addressing (examples: @A, @IX, @EP)
A	Accumulator (8 or 16 bits, which are determined depending on the instruction being used)
AH	Higher 8 bits of the accumulator (8 bits)
AL	Lower 8 bits of the accumulator (8 bits)
T	Temporary accumulator (8 or 16 bits, which are determined depending on the instruction being used)
TH	Higher 8 bits of the temporary accumulator (8 bits)
TL	Lower 8 bits of the temporary accumulator (8 bits)
IX	Index register (16 bits)
EP	Extra pointer (16 bits)
PC	Program counter (16 bits)
SP	Stack pointer (16 bits)
PS	Program status (16 bits)
dr	Either accumulator or index register (16 bits)
CCR	Condition code register (8 bits)
RP	Register bank pointer (5 bits)
Ri	General-purpose register (8 bits, i = 0 to 7)
X	X is immediate data (8 or 16 bits, which are determined depending on the instruction being used).
(X)	The content of X is to be accessed (8 or 16 bits, which are determined depending on the instruction being used).
((X))	The address indicated by the X is to be accessed (8 or 16 bits, which are determined depending on the instruction being used).

B.2 Addressing

The F²MC-8L has the following ten addressing modes:

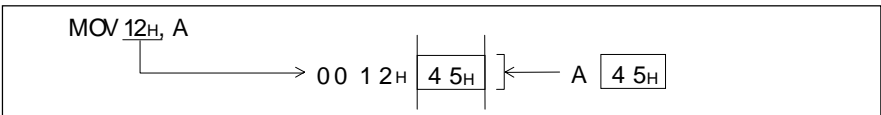
- Direct addressing
- Extended addressing
- Bit direct addressing
- Index addressing
- Pointer addressing
- General-purpose register addressing
- Immediate addressing
- Vector addressing
- Relative addressing
- Inherent addressing

■ Explanation of Addressing

○ Direct addressing

Direct addressing is indicated by `dir` in the instruction list. This addressing is used to access the area between 0000_H and 00FF_H. In this addressing mode, the higher byte of the address is 00_H and the lower byte is specified by the operand. Figure B.2-1 "Example of Direct Addressing" shows an example.

Figure B.2-1 Example of Direct Addressing

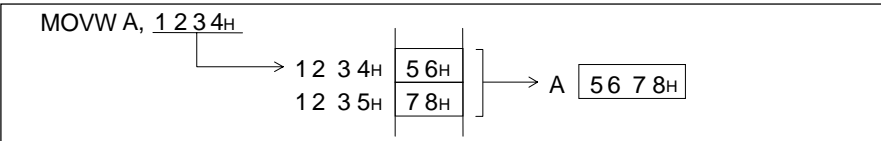


○ Extended addressing

Extended addressing is indicated by `ext` in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the first operand specifies the higher byte of the address, and the second operand specifies the lower byte.

Figure B.2-2 "Example of Extended Addressing" shows an example.

Figure B.2-2 Example of Extended Addressing

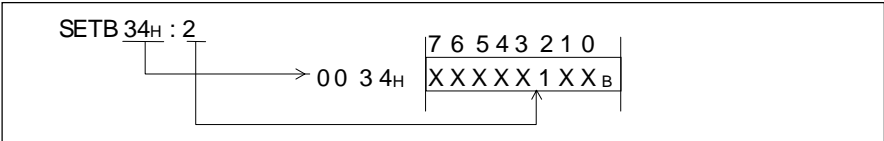


○ Bit direct addressing

Bit direct addressing is indicated by `dir:b` in the instruction list. This addressing is used to access a particular bit in the area between `0000H` and `00FFH`. In this addressing mode, the higher byte of the address is `00H` and the lower byte is specified by the operand. The bit position at the address is specified by the lower three bits of the operation code.

Figure B.2-3 "Example of Bit Direct Addressing" shows an example.

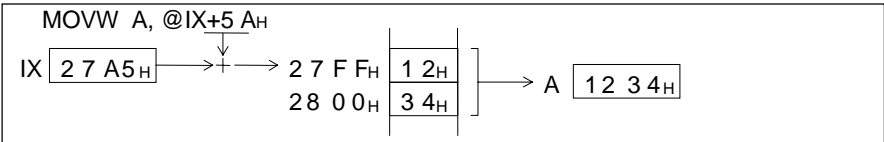
Figure B.2-3 Example of Bit Direct Addressing



○ Index addressing

Index addressing is indicated by `@IX+off` in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the address is the value resulting from sign-extending the contents of the first operand and adding them to `IX` (index register). Figure B.2-4 "Example of Index Addressing" shows an example.

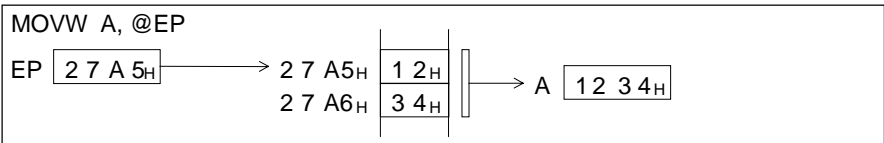
Figure B.2-4 Example of Index Addressing



○ Pointer addressing

Pointer addressing is indicated by `@EP` in the instruction list. This addressing is used to access the entire 64-KB area. In this addressing mode, the address is contained in `EP` (extra pointer). Figure B.2-5 "Example of Pointer Addressing" shows an example.

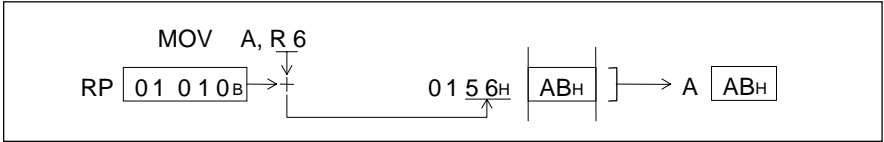
Figure B.2-5 Example of Pointer Addressing



○ General-purpose register addressing

General-purpose register addressing is indicated by `Ri` in the instruction list. This addressing is used to access a register bank in the general-purpose register area. In this addressing mode, the higher byte of the address is always `01` and the lower byte is specified based on the contents of `RP` (register bank pointer) and the lower three bits of the operation code. Figure B.2-6 "Example of general-purpose register addressing" shows an example.

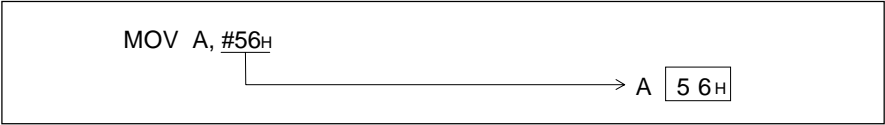
Figure B.2-6 Example of General-purpose Register Addressing



○ Immediate addressing

Immediate addressing is indicated by #d8 in the instruction list. This addressing is used when immediate data is required. In this addressing mode, the operand is used as immediate data. Whether the data is specified in bytes or words is determined by the operation code. Figure B.2-7 "Example of Immediate Addressing" shows an example.

Figure B.2-7 Example of Immediate Addressing



○ Vector addressing

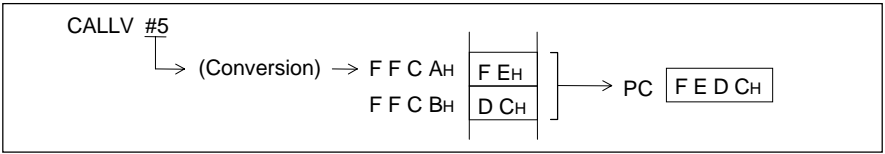
Vector addressing is indicated by vct in the instruction list. This addressing is used to branch to a subroutine address stored in the vector table. In this addressing mode, vct information is contained in the operation codes, and the corresponding table addresses are created as shown in Table B.2-1 "Vector Table Addresses Corresponding to vct".

Table B.2-1 Vector Table Addresses Corresponding to vct

#vct	Vector table address (higher address:lower address of branch destination)
0	FFC0 _H : FFC1 _H
1	FFC2 _H : FFC3 _H
2	FFC4 _H : FFC5 _H
3	FFC6 _H : FFC7 _H
4	FFC8 _H : FFC9 _H
5	FFCA _H : FF CB _H
6	FFCC _H : FF CD _H
7	FFCE _H : FF CF _H

Figure B.2-8 "Example of Vector Addressing" shows an example.

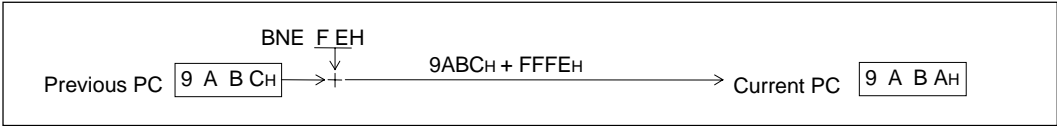
Figure B.2-8 Example of Vector Addressing



○ Relative addressing

Relative addressing is indicated by rel in the instruction list. This addressing is used to branch to within the area between the address 128 bytes higher and that 128 bytes lower relative to the address contained in the PC (program counter). In this addressing mode, the result of a signed addition of the contents of the operand to the PC is stored in the PC. Figure B.2-9 "Example of Relative Addressing" shows an example.

Figure B.2-9 Example of Relative Addressing

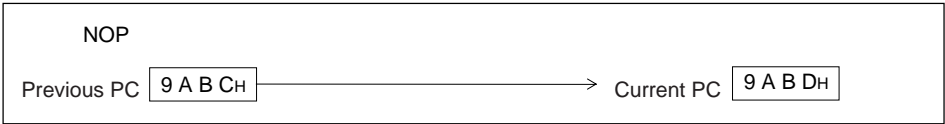


In this example, a branch to the address of the BNE operation code occurs, thus resulting in an infinite loop.

○ Inherent addressing

Inherent addressing is indicated as the addressing without operands in the instruction list. This addressing is used to perform the operation determined by the operation code. In this addressing mode, different operations are performed via different instructions. Figure B.2-10 "Example of Inherent Addressing" shows an example.

Figure B.2-10 Example of Inherent Addressing



B.3 Special Instructions

This section describes the special instructions used for other than addressing.

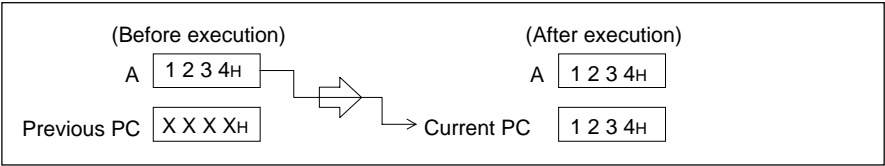
■ Special Instructions

○ JMP @A

This instruction sets the contents of A (accumulator) to PC (program counter) as the address, and causes a branch to that address. One of the N branch destination addresses is selected from a table, and then transferred to A. The instruction can be executed to perform N-branch processing.

Figure B.3-1 "JMP @A" shows a summary of the instruction.

Figure B.3-1 JMP @A

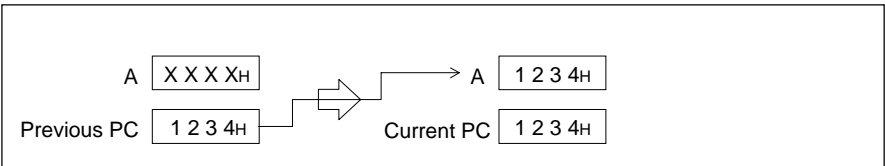


○ MOVW A, PC

This instruction performs the operation which is the reverse of that performed by JMP @A. That is, the instruction stores the contents of PC in A. When the instruction is executed in the main routine, so that a specific subroutine is called, whether A contains a predetermined value can be checked by the subroutine. This can be used to determine that the branch source is not any unexpected section of the program and to check for program runaway.

Figure B.3-2 "MOVW A, PC" shows a summary of the instruction.

Figure B.3-2 MOVW A, PC



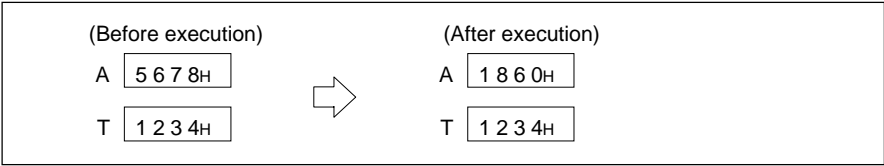
After the MOVW A, PC instruction is executed, A contains the address of the operation code of the next instruction, rather than the address of the operation code of MOVW A, PC. Accordingly, Figure B.3-2 "MOVW A, PC" shows that A contains 1234_H, which is the address of the operation code of the instruction that follows MOVW A, PC.

MULU A

This instruction performs an unsigned multiplication of AL (lower eight bits of the accumulator) and TL (lower eight bits 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 eight bits of the accumulator) and TH (higher eight bits 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 B.3-3 "MULU" shows a summary of the instruction.

Figure B.3-3 MULU



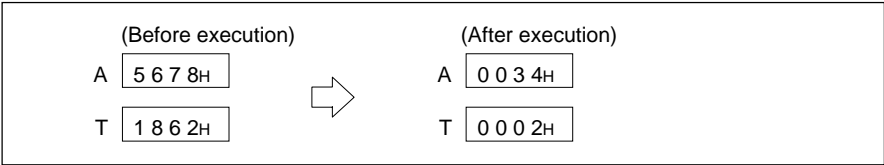
DIVU A

This instruction divides the 16-bit value in T by the unsigned 8-bit value in AL, and stores the 8-bit result and the 8-bit remainder in AL and TL, respectively. A value of 0 is set to both AH and TH. The contents of AH before execution of the instruction are not used for the operation. An unpredictable result is produced from data that results in more than eight bits. In addition, there is no indication of the result having more than eight bits. Therefore, if it is likely that data will cause a result of more than eight bits, the data must be checked to ensure that the result will not have more than eight bits before it is used.

The instruction does not change the flags, and therefore care must be taken when a branch may occur depending on the result of a division.

Figure B.3-4 "DIVU A" shows a summary of the instruction.

Figure B.3-4 DIVU A

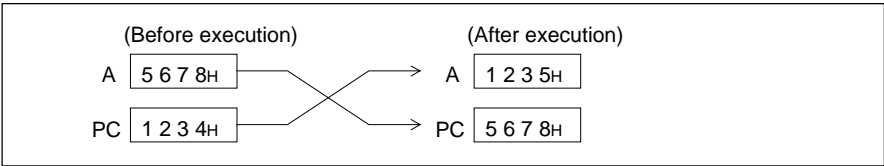


XCHW A, PC

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 contains the address that follows the address of the operation code of MOVW A, PC. This instruction is effective especially when it is used in the main routine to specify a table for use in a subroutine.

Figure B.3-5 "XCHW A, PC" shows a summary of the instruction.

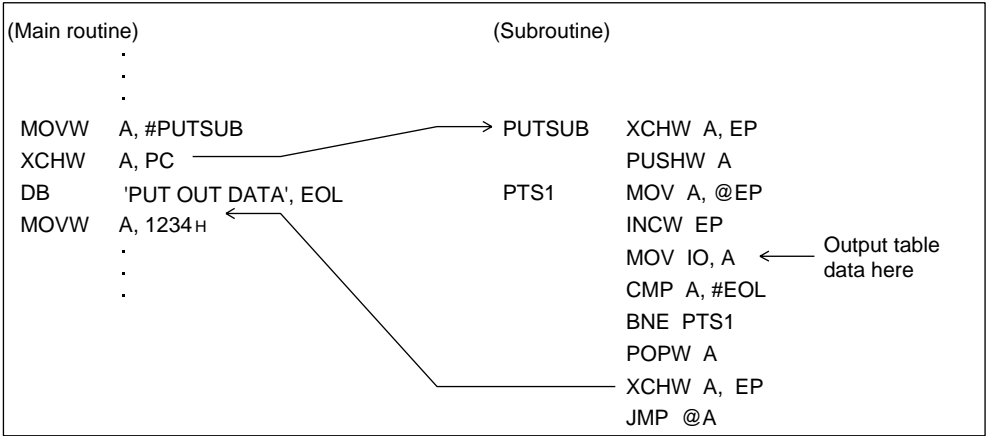
Figure B.3-5 XCHW A, PC



After the XCHW A, PC instruction is executed, A contains the address of the operation code of the next instruction, rather than the address of the operation code of XCHW A, PC. Accordingly, Figure B.3-5 "XCHW A, PC" shows that A contains 1235_H, which is the address of the operation code of the instruction that follows XCHW A, PC. This is why 1235_H is stored instead of 1234_H.

Figure B.3-6 "Example of Using XCHW A, PC" shows an assembly language example.

Figure B.3-6 Example of Using XCHW A, PC

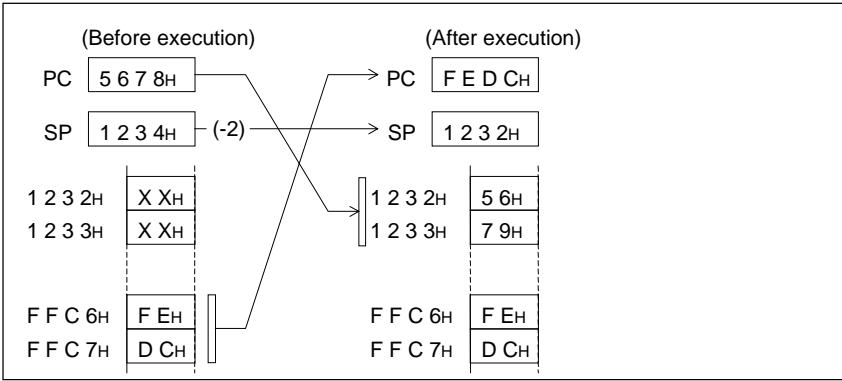


○ CALLV #vct

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 B.3-7 "Example of Executing CALLV #3" shows a summary of the instruction.

Figure B.3-7 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 B.3-7 "Example of Executing CALLV #3" shows that the value saved in the stack (1232_H and 1233_H) is 5679_H, which is the address of the operation code of the instruction that follows CALLV #vct (return address).

B.4 Bit Manipulation Instructions (SETB, CLRB)

Some bits of peripheral function registers include bits that are read by a bit manipulation instruction differently than usual.

■ Read-modify-write Operation

By using these bit manipulation instructions, only the specified bit in a register or RAM location can be set to 1 (SETB) or cleared to 0 (CLRB). However, as the CPU operates on 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 B.4-1 "Bus Operation for Bit Manipulation Instructions" shows bus operation for bit manipulation instructions.

Table B.4-1 Bus operation for Bit Manipulation Instructions

CODE	MNEMONIC	TO	Cycle	Address bus	Data bus	\overline{RD}	\overline{WR}	RMW
A0 to A7	CLRB dir:b	4	1	N+1	dir	0	1	0
			2	dir address	Data	0	1	1
A8 to AF	SETB dir:b		3	dir address	Data	1	0	0
			4	N+2	Next instruction	0	1	0

■ Read Operation Upon the Execution of Bit Manipulation Instructions

For some I/O ports and for the interrupt request flag bits, the value to be read 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 an output latch value is read during a bit manipulation. This prevents the other output latch 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 a value of 0 which would otherwise be written to the interrupt request flag bit when another bit is manipulated.

B.5 F²MC-8L Instructions

Table B.5-1 "Transfer Instructions" to Table B.5-4 "Other Instructions" list the instructions used with the F²MC-8L.

■ Transfer Instructions

Table B.5-1 Transfer Instructions (1 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	MOV dir, A	3	2	(dir)<--(A)	-	-	-	-	-	-	-	45
2	MOV @IX+off, A	4	2	((IX)+off)<--(A)	-	-	-	-	-	-	-	46
3	MOV ext, A	4	3	(ext)<--(A)	-	-	-	-	-	-	-	61
4	MOV @EP, A	3	1	((EP))<--(A)	-	-	-	-	-	-	-	47
5	MOV Ri, A	3	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	4	2	(A)<--((IX)+off)	AL	-	-	+	+	-	-	06
9	MOV A, ext	4	3	(A)<--(ext)	AL	-	-	+	+	-	-	60
10	MOV A, @A	3	1	(A)<--((A))	AL	-	-	+	+	-	-	92
11	MOV A, @EP	3	1	(A)<--((EP))	AL	-	-	+	+	-	-	07
12	MOV A, Ri	3	1	(A)<--(Ri)	AL	-	-	+	+	-	-	08 to 0F
13	MOV dir, #d8	4	3	(dir)<--d8	-	-	-	-	-	-	-	85
14	MOV @IX+off, #d8	5	3	((IX)+off)<--d8	-	-	-	-	-	-	-	86
15	MOV @EP, #d8	4	2	((EP))<--d8	-	-	-	-	-	-	-	87
16	MOV Ri, #d8	4	2	(Ri)<--d8	-	-	-	-	-	-	-	88 to 8F
17	MOVW dir, A	4	2	(dir)<--(AH), (dir+1)<--(AL)	-	-	-	-	-	-	-	D5
18	MOVW @IX+off, A	5	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	4	1	((EP))<--(AH), ((EP)+1)<--(AL)	-	-	-	-	-	-	-	D7
21	MOVW EP, A	2	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

APPENDIX B Overview of Instructions

Table B.5-1 Transfer Instructions (2 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
24	MOVW A, @IX+off	5	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	4	1	(AH)<--((A)), (AL)<--((A)+1)	AL	AH	dH	+	+	-	-	93
27	MOVW A, @EP	4	1	(AH)<--((EP)), (AL)<--((EP)+1)	AL	AH	dH	+	+	-	-	C7
28	MOVW A, EP	2	1	(A)<--(EP)	-	-	dH	-	-	-	-	F3
29	MOVW EP, #d16	3	3	(EP)<--d16	-	-	-	-	-	-	-	E7
30	MOVW IX, A	2	1	(IX)<--(A)	-	-	-	-	-	-	-	E2
31	MOVW A, IX	2	1	(A)<--(IX)	-	-	dH	-	-	-	-	F2
32	MOVW SP, A	2	1	(SP)<--(A)	-	-	-	-	-	-	-	E1
33	MOVW A, SP	2	1	(A)<--(SP)	-	-	dH	-	-	-	-	F1
34	MOV @A, T	3	1	((A))<--(T)	-	-	-	-	-	-	-	82
35	MOVW @A, T	4	1	((A))<--(TH), ((A)+1)<--(TL)	-	-	-	-	-	-	-	83
36	MOVW IX, #d16	3	3	(IX)<--d16	-	-	-	-	-	-	-	E6
37	MOVW A, PS	2	1	(A)<--(PS)	-	-	dH	-	-	-	-	70
38	MOVW PS, A	2	1	(PS)<--(A)	-	-	-	+	+	+	+	71
39	MOVW SP, #d16	3	3	(SP)<--d16	-	-	-	-	-	-	-	E5
40	SWAP	2	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	2	1	(AL)<-- -->(TL)	AL	-	-	-	-	-	-	42
44	XCHW A, T	3	1	(A)<-- -->(T)	AL	AH	dH	-	-	-	-	43
45	XCHW A, EP	3	1	(A)<-- -->(EP)	-	-	dH	-	-	-	-	F7
46	XCHW A, IX	3	1	(A)<-- -->(IX)	-	-	dH	-	-	-	-	F6
47	XCHW A, SP	3	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 two or more operands, they are assumed to be saved in the order indicated by MNEMONIC.

Arithmetic Instructions

Table B.5-2 Arithmetic Operation Instructions (1 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	ADDC A, Ri	3	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	4	2	$(A) \leftarrow (A) + ((IX) + off) + C$	-	-	-	+	+	+	+	26
5	ADDC A, @EP	3	1	$(A) \leftarrow (A) + ((EP)) + C$	-	-	-	+	+	+	+	27
6	ADDCW A	3	1	$(A) \leftarrow (A) + (T) + C$	-	-	dH	+	+	+	+	23
7	ADDC A	2	1	$(AL) \leftarrow (AL) + (TL) + C$	-	-	-	+	+	+	+	22
8	SUBC A, Ri	3	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	4	2	$(A) \leftarrow (A) - ((IX) + off) - C$	-	-	-	+	+	+	+	36
12	SUBC A, @EP	3	1	$(A) \leftarrow (A) - ((EP)) - C$	-	-	-	+	+	+	+	37
13	SUBCW A	3	1	$(A) \leftarrow (T) - (A) - C$	-	-	dH	+	+	+	+	33
14	SUBC A	2	1	$(AL) \leftarrow (TL) - (AL) - C$	-	-	-	+	+	+	+	32
15	INC Ri	4	1	$(Ri) \leftarrow (Ri) + 1$	-	-	-	+	+	+	-	C8 to CF
16	INCW EP	3	1	$(EP) \leftarrow (EP) + 1$	-	-	-	-	-	-	-	C3
17	INCW IX	3	1	$(IX) \leftarrow (IX) + 1$	-	-	-	-	-	-	-	C2
18	INCW A	3	1	$(A) \leftarrow (A) + 1$	-	-	dH	+	+	-	-	C0
19	DEC Ri	4	1	$(Ri) \leftarrow (Ri) - 1$	-	-	-	+	+	+	-	D8 to DF
20	DECW EP	3	1	$(EP) \leftarrow (EP) - 1$	-	-	-	-	-	-	-	D3
21	DECW IX	3	1	$(IX) \leftarrow (IX) - 1$	-	-	-	-	-	-	-	D2
22	DECW A	3	1	$(A) \leftarrow (A) - 1$	-	-	dH	+	+	-	-	D0
23	MULU A	19	1	$(A) \leftarrow (AL) \times (TL)$	-	-	dH	-	-	-	-	01
24	DIVU A	21	1	$(A) \leftarrow (T) / (AL), \text{MOD} \rightarrow (T)$	dL	00	00	-	-	-	-	11
25	ANDW A	3	1	$(A) \leftarrow (A) \wedge (T)$	-	-	dH	+	+	R	-	63
26	ORW A	3	1	$(A) \leftarrow (A) \vee (T)$	-	-	dH	+	+	R	-	73
27	XORW A	3	1	$(A) \leftarrow (A) \vee (T)$	-	-	dH	+	+	R	-	53
28	CMP A	2	1	$(TL) - (AL)$	-	-	-	+	+	+	+	12
29	CMPW A	3	1	$(T) - (A)$	-	-	-	+	+	+	+	13
30	RORC A	2	1	$\boxed{\rightarrow C \rightarrow A}$	-	-	-	+	+	-	+	03

APPENDIX B Overview of Instructions

Table B.5-2 Arithmetic Operation Instructions (2 / 2)

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
31	ROL A	2	1	$C \leftarrow A \ll 1$	-	-	-	+	+	-	+	02
32	CMP A, #d8	2	2	(A)-d8	-	-	-	+	+	+	+	14
33	CMP A, dir	3	2	(A)-(dir)	-	-	-	+	+	+	+	15
34	CMP A, @EP	3	1	(A)-((EP))	-	-	-	+	+	+	+	17
35	CMP A, @IX+off	4	2	(A)-((IX)+off)	-	-	-	+	+	+	+	16
36	CMP A, Ri	3	1	(A)-(Ri)	-	-	-	+	+	+	+	18 to 1F
37	DAA	2	1	decimal adjust for addition	-	-	-	+	+	+	+	84
38	DAS	2	1	decimal adjust for subtraction	-	-	-	+	+	+	+	94
39	XOR A	2	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	3	1	(A) \leftarrow -(AL) \vee ((EP))	-	-	-	+	+	R	-	57
43	XOR A, @IX+off	4	2	(A) \leftarrow -(AL) \vee ((IX)+off)	-	-	-	+	+	R	-	56
44	XOR A, Ri	3	1	(A) \leftarrow -(AL) \vee (Ri)	-	-	-	+	+	R	-	58 to 5F
45	AND A	2	1	(A) \leftarrow -(AL) \wedge (TL)	-	-	-	+	+	R	-	62
46	AND A, #d8	2	2	(A) \leftarrow -(AL) \wedge d8	-	-	-	+	+	R	-	64
47	AND A, dir	3	2	(A) \leftarrow -(AL) \wedge (dir)	-	-	-	+	+	R	-	65
48	AND A, @EP	3	1	(A) \leftarrow -(AL) \wedge ((EP))	-	-	-	+	+	R	-	67
49	AND A, @IX+off	4	2	(A) \leftarrow -(AL) \wedge ((IX)+off)	-	-	-	+	+	R	-	66
50	AND A, Ri	3	1	(A) \leftarrow -(AL) \wedge (Ri)	-	-	-	+	+	R	-	68 to 6F
51	OR A	2	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 (dir)	-	-	-	+	+	R	-	75
54	OR A, @EP	3	1	(A) \leftarrow -(AL) \vee ((EP))	-	-	-	+	+	R	-	77
55	OR A, @IX+off	4	2	(A) \leftarrow -(AL) \vee ((IX)+off)	-	-	-	+	+	R	-	76
56	OR A, Ri	3	1	(A) \leftarrow -(AL) \vee (Ri)	-	-	-	+	+	R	-	78 to 7F
57	CMP dir, #d8	5	3	(dir)-d8	-	-	-	+	+	+	+	95
58	CMP @EP, #d8	4	2	((EP))-d8	-	-	-	+	+	+	+	97
59	CMP @IX+off, #d8	5	3	((IX)+off)-d8	-	-	-	+	+	+	+	96
60	CMP Ri, #d8	4	2	(Ri)-d8	-	-	-	+	+	+	+	98 to 9F
61	INCW SP	3	1	(SP) \leftarrow -(SP)+1	-	-	-	-	-	-	-	C1
62	DECW SP	3	1	(SP) \leftarrow -(SP)-1	-	-	-	-	-	-	-	D1

■ Branch Instructions

Table B.5-3 Branch Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	BZ/BEQ rel	3	2	if Z=1 then PC←--PC+rel	-	-	-	-	-	-	-	FD
2	BNZ/BNE rel	3	2	if Z=0 then PC←--PC+rel	-	-	-	-	-	-	-	FC
3	BC/BLO rel	3	2	if C=1 then PC←--PC+rel	-	-	-	-	-	-	-	F9
4	BNC/BHS rel	3	2	if C=0 then PC←--PC+rel	-	-	-	-	-	-	-	F8
5	BN rel	3	2	if N=1 then PC←--PC+rel	-	-	-	-	-	-	-	FB
6	BP rel	3	2	if N=0 then PC←--PC+rel	-	-	-	-	-	-	-	FA
7	BLT rel	3	2	if V ∨ N=1 then PC←--PC+rel	-	-	-	-	-	-	-	FF
8	BGE rel	3	2	if V ∨ N=0 then PC←--PC+rel	-	-	-	-	-	-	-	FE
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	2	1	(PC)←--(A)	-	-	-	-	-	-	-	E0
12	JMP ext	3	3	(PC)←--ext	-	-	-	-	-	-	-	21
13	CALLV #vct	6	1	vector call	-	-	-	-	-	-	-	E8 to EF
14	CALL ext	6	3	subroutine call	-	-	-	-	-	-	-	31
15	XCHW A, PC	3	1	(PC)←--(A), (A)←--(PC)+1	-	-	dH	-	-	-	-	F4
16	RET	4	1	return from subroutine	-	-	-	-	-	-	-	20
17	RETI	6	1	return from interrupt	-	-	-	restore				30

APPENDIX B Overview of Instructions

■ Other Instructions

Table B.5-4 Other Instructions

No.	MNEMONIC	~	#	Operation	TL	TH	AH	N	Z	V	C	OP CODE
1	PUSHW A	4	1	((SP))<--(A), (SP)<--(SP)-2	-	-	-	-	-	-	-	40
2	POPW A	4	1	(A)<--((SP)), (SP)<--(SP)+2	-	-	dH	-	-	-	-	50
3	PUSHW IX	4	1	((SP))<--(IX), (SP)<--(SP)-2	-	-	-	-	-	-	-	41
4	POPW IX	4	1	(IX)<--((SP)), (SP)<--(SP)+2	-	-	-	-	-	-	-	51
5	NOP	1	1	No operation	-	-	-	-	-	-	-	00
6	CLRC	1	1	(C)<--0	-	-	-	-	-	-	R	81
7	SETC	1	1	(C)<--1	-	-	-	-	-	-	S	91
8	CLRI	1	1	(I)<--0	-	-	-	-	-	-	-	80
9	SETI	1	1	(I)<--1	-	-	-	-	-	-	-	90

B.6 Instruction Map

Table B.6-1 "F²MC-8L Instruction Map" shows the F²MC-8L instruction map.

■ Instruction Map

Table B.6-1 F²MC-8L Instruction Map

H L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SWAP	RET	RETI	PUSHW A	POPW A	MOV A, ext	MOVW A, PS	CLRI	SETI	CLRB dir: 0, rel	BBC dir: 0, rel	INCW	DECW A	JMP @A	MOVW A, PC
1	MULLU A	DIVU A	JMP addr16	CALL addr16	PUSHW IX	POPW IX	MOV ext, A	MOVW PS, A	CLRC	SETC	CLRB dir: 1, rel	BBC dir: 1, rel	INCW	DECW SP	MOVW SP, A	MOVW A, SP
2	ROLC A	CMP A	ADDC A	SUBC A	XCH A, T	XOR A, T	AND A	OR A	MOV @A, T	MOV A, @A	CLRB dir: 2, rel	BBC dir: 2, rel	INCW	DECW IX	MOVW IX, A	MOVW A, IX
3	RORC A	CMPW A	ADDCW A	SUBCW A	XCHW A, T	XORW A, T	ANDW A	ORW A	MOVW @A, T	MOVW A, @A	CLRB dir: 3, rel	BBC dir: 3, rel	INCW	DECW EP	MOVW EP, A	MOVW A, EP
4	MOV A, #d8	CMP A, #d8	ADDC A, #d8	SUBC A, #d8	XCHW A, #d8	XOR A, #d8	AND A, #d8	OR A, #d8	DAA	DAS	CLRB dir: 4, rel	BBC dir: 4, rel	MOVW A, ext	MOVW ext, A	MOVW A, #d16	MOVW A, PC
5	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, #d8	CMP dir, #d8	CLRB dir: 5, rel	BBC dir: 5, rel	MOVW A, dir	MOVW dir, A	MOVW SP, #d16	MOVW A, SP
6	MOV A, @IX+d	CMP A, @IX+d	ADDC A, @IX+d	SUBC A, @IX+d	MOV @IX+d, A	XOR A, @IX+d	AND A, @IX+d	OR A, @IX+d	MOV @IX+d, #d8	CMP @IX+d, #d8	CLRB dir: 6, rel	BBC dir: 6, rel	MOVW A, @IX+d	MOVW @IX+d, A	MOVW IX, #d16	MOVW A, IX
7	MOV A, @EP	CMP A, @EP	ADDC A, @EP	SUBC A, @EP	MOV @EP, A	XOR A, @EP	AND A, @EP	OR A, @EP	MOV @EP, d8	CMP @EP, d8	CLRB dir: 7, rel	BBC dir: 7, rel	MOVW A, @EP	MOVW @EP, A	MOVW EP, #d16	MOVW A, EP
8	MOV A, R0	CMP A, R0	ADDC A, R0	SUBC A, R0	MOV R0, A	XOR A, R0	AND A, R0	OR A, R0	MOV R0, #d8	CMP R0, #d8	SETB dir: 0, rel	BBS dir: 0, rel	INC	DEC	CALLV #0	rel
9	MOV A, R1	CMP A, R1	ADDC A, R1	SUBC A, R1	MOV R1, A	XOR A, R1	AND A, R1	OR A, R1	MOV R1, #d8	CMP R1, #d8	SETB dir: 1, rel	BBS dir: 1, rel	INC	DEC	CALLV #1	rel
A	MOV A, R2	CMP A, R2	ADDC A, R2	SUBC A, R2	MOV R2, A	XOR A, R2	AND A, R2	OR A, R2	MOV R2, #d8	CMP R2, #d8	SETB dir: 2, rel	BBS dir: 2, rel	INC	DEC	CALLV #2	rel
B	MOV A, R3	CMP A, R3	ADDC A, R3	SUBC A, R3	MOV R3, A	XOR A, R3	AND A, R3	OR A, R3	MOV R3, #d8	CMP R3, #d8	SETB dir: 3, rel	BBS dir: 3, rel	INC	DEC	CALLV #3	rel
C	MOV A, R4	CMP A, R4	ADDC A, R4	SUBC A, R4	MOV R4, A	XOR A, R4	AND A, R4	OR A, R4	MOV R4, #d8	CMP R4, #d8	SETB dir: 4, rel	BBS dir: 4, rel	INC	DEC	CALLV #4	rel
D	MOV A, R5	CMP A, R5	ADDC A, R5	SUBC A, R5	MOV R5, A	XOR A, R5	AND A, R5	OR A, R5	MOV R5, #d8	CMP R5, #d8	SETB dir: 5, rel	BBS dir: 5, rel	INC	DEC	CALLV #5	rel
E	MOV A, R6	CMP A, R6	ADDC A, R6	SUBC A, R6	MOV R6, A	XOR A, R6	AND A, R6	OR A, R6	MOV R6, #d8	CMP R6, #d8	SETB dir: 6, rel	BBS dir: 6, rel	INC	DEC	CALLV #6	rel
F	MOV A, R7	CMP A, R7	ADDC A, R7	SUBC A, R7	MOV R7, A	XOR A, R7	AND A, R7	OR A, R7	MOV R7, #d8	CMP R7, #d8	SETB dir: 7, rel	BBS dir: 7, rel	INC	DEC	CALLV #7	rel

APPENDIX C Write Specifications for One-Time PROM and EPROM Microcomputer

The MB89P585B/BW and MB89P589B have a PROM mode and can be programmed using a general-purpose ROM programmer via a dedicated adaptor. Note that the electronic signature mode is not available.

■ ROM Programmer Adapters and Recommended ROM Programmers

Table C-1 "ROM Programmer Adaptors and Recommended ROM Programmers" lists ROM programmer adaptors and recommended ROM programmers.

Table C-1 ROM Programmer Adaptors and Recommended ROM Programmers

Package name	Compatible adaptor	Recommended programmer maker and the programmer name
	Sun Hayato Co., Ltd.	Ando Electric Co., Ltd.
FPT-64P-M03	ROM2-64LQF-32DP-8LA	AF9708 (ver1.40 or more) AF9709 (ver1.40 or more) AF9723 (ver1.50 or more)
FPT-64P-M09	ROM2-64QF2-32DP-8LA2	

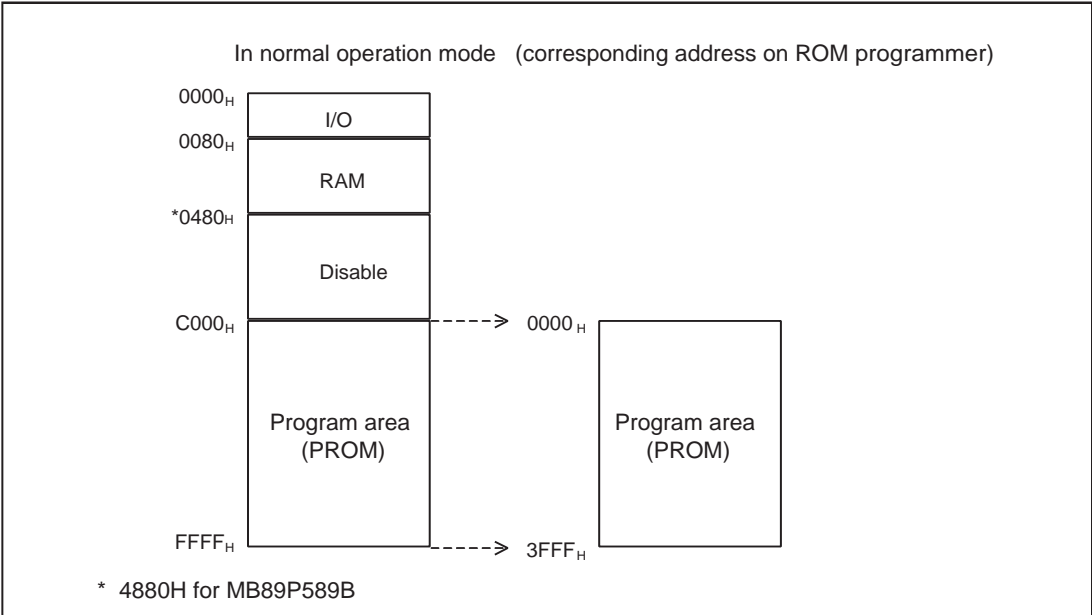
Inquiries:

Sun Hayato Co., Ltd. : Phone (81)-3-3986-0403

Ando Electric Co.,Ltd. : Phone (81)-44-549-7300

■ Memory Map in PROM Mode

Figure C-1 Memory Map in PROM Mode



■ Programming to EPROM (When using the Ando Electric Co.,Ltd. programmer)

1. Set the EPROM programmer type code to 17209.
2. Load program data on to the area of 0000_H to 3FFF_H of the EPROM programmer.
3. Write to the area of C000_H to FFFF_H by the EPROM programmer.

■ Programming Yield

All bits cannot be programmed at Fujitsu shipping test to a blanked OTPROM microcomputer, due to its nature. For this reason, a programming yield of 100% cannot be assured at all times.

APPENDIX D Pin State of MB89580B/BW Series

This section describes the pin state for each operation of the MB89580B/BW series.

■ Pin State for Each Operation

Table D-1 Pin States in Each Mode

Pin name	During normal operation	In sleep mode	In stop mode SPL=0	In stop mode SPL=1	During reset ^{*3}
P30/ $\overline{\text{INT0}}$ /CLK to P37/ $\overline{\text{INT7}}$	Port input and output/External interrupt input	Hold/External interrupt input	Hold/External interrupt input	Hi-Z/External interrupt input	Hi-Z
X0	Oscillation input	Oscillation input	Hi-Z	Hi-Z	Oscillation input
X1	Oscillation output	Oscillation output	"H" output	"H" output	Oscillation output
MOD0,1	Mode input	Mode input	Mode input	Mode input	Mode input
$\overline{\text{RST}}$	Reset input and output	Reset input and output	Reset input and output	Reset input and output	Reset input and output
P00 to P07	Port input and output	Hold			
P10 to P17					
P20 to P27	Port output				
P40/D0 to P43/D3	Port input and output/Resource input and output	Hold/Resource input and output	Hold	Hi-Z	Hi-Z
P44/UCK/D4					
P45/UO/D5					
P46/UI/PWM1/D6					
P47/PWM2/D7					
P50/OBF/IBFX/W					
P51/R					
P52/EFX					
P53/AO/FFX					
P54/CEX					
P60 to P67	Port input/Resource input	Port input/Resource input	Port input/Resource input	Port input/Resource input	Port input/ Resource input
RPVP	USB port input and output	USB port input and output	USB port input ^{*1}	USB port input ^{*1}	Low level output/ Hi-Z ^{*2}
RPVM					

Hi-Z: Denotes high impedance

SPL: Pin state selection bit of standby control register (STBC)

Hold: Pin set for output holds the pin state (level) just before the mode switching.

^{*1}: Port input when the system stops because of the USB port status; The latest state is retained when the system stops otherwise.

^{*2}: Low level output for the MB89580BW series; Hi-Z for the MB89580B series.

^{*3}: The P60 to P67 become indeterminate outputs during a period of power-on reset.

INDEX

**The index follows on the next page.
This is listed in alphabetic order.**

Index

Numerics

16-Bit Data	
Storage State of 16-Bit Data in RAM	26
16-Bit Operand	
Storage State of a 16-Bit Operand	26
8-Bit PWM Timer	
8-Bit PWM Timer 1 (CH1) and 8-Bit PWM	
Timer 2 (CH2)	147
Block Diagram of Pins Related to the 2-Channel	
8-Bit PWM Timer	149
Block Diagram of the 2-Channel 8-Bit PWM Timer	
.....	146
Notes on Using the 2-Channel 8-Bit PWM Timer	
.....	174
Overview of the 2-Channel 8-Bit PWM Timer	
(PWM Timer Function)	143
Pins Related to the 2-Channel 8-Bit PWM Timer	
.....	149
8-bit Timer Mode	
Program Example of 8-bit Timer Mode	175

A

Addressing	
Explanation of Addressing	315
Arithmetic Operation	
Bits That Indicate Arithmetic Operation Results	
.....	30

B

Block Diagram	
Block Diagram of External Interrupt Circuit....	185
Block Diagram of Parallel Port	201
Block Diagram of Pins Related to the 2-Channel	
8-Bit PWM Timer	149
Block Diagram of Port 5	108
Block Diagram of Port 6	115
Block Diagram of the 2-Channel 8-Bit PWM Timer	
.....	146
Block Diagram of the Clock Controller	55
Block Diagram of the External Reset Pin	48
Block Diagram of the Pins Related to Clock	
Output	301
Block Diagram of the Timebase Timer	122
Block Diagram of the Watchdog Timer	133
Block Diagram of UART/SIO	269
Block Diagram of USB Function Circuit	216
Full Block Diagram of MB89580B/BW Series....	8
Bus Reset	
USB Bus Reset	257

C

CCR	
Configuration of Condition Code Register (CCR)	
.....	30
CH12PWM	
CH12PWM Mode Operation	168
CH12PWM Output Control Circuit	147
CK12 Selector	
CK12 Selector	147
CKR	
Register Related to Clock Output Control	
Register (CKR)	302
CLK	
Pins Related to Clock Output (CLK)	301
Clock	
Operation of Clock Supply Function	127
Clock Controller	
Block Diagram of the Clock Controller	55
Clock Generator	
Clock Generator	54
Clock Mode	
Clock Mode Operating State	59
Clock Output	
Block Diagram of the Pins Related to Clock	
Output	301
Clock Output Function	300
Pins Related to Clock Output (CLK)	301
Clock Output Control Register	
Register Related to Clock Output Control	
Register (CKR)	302
Clock Supply Function	
Operation of Clock Supply Function	127
Clock Supply Map	
Clock Supply Map	52
CNTR	
PWM Control Register 1,2,3	
(CNTR1,CNTR2,CNTR3)	147, 150
CNTR1	
PWM Control Register 1 (CNTR1)	151
CNTR2	
PWM Control Register 2 (CNTR2)	153
CNTR3	
PWM Control Register 3 (CNTR3)	155
Command Responses	
Operations of Each Register for Command	
Responses	258

COMR		EndPoint Enable Register	
PWM Compare Register 1,2 (COMR1,COMR2)	147	EndPoint Enable Register (EPER).....	241
COMR1		EndPoint Setup Registers	
PWM Compare Register 1 (COMR1)	157	EndPoint Setup Registers	
COMR2		(EPBR0,EPBRn1,EPBRn2)	242
PWM Compare Register 2 (COMR2)	159	EPBR	
Condition Code Register		EndPoint Setup Registers	
Configuration of Condition Code Register (CCR)	30	(EPBR0,EPBRn1,EPBRn2)	242
Connection Detection		EPER	
Connection Detection	256	EndPoint Enable Register (EPER).....	241
D		EPROM	
DBAR		Programming to EPROM (When using the Ando	
DMA Base Address Register (DBAR)	225	Electric Co.,Ltd. programmer)	331
DMA Base Address Register Low (DBAR)....	227	External FIFO	
DBARH		DMA Transfer to or from External FIFO	254
DMA Base Address Register High (DBARH)	227	External Interrupt	
DDR0		Function of External Interrupt Circuit (Level	
Port 0 Registers (PDR0 and DDR0)	80	Detection)	184
DDR1		External Interrupt Circuit	
Port 1 Registers (PDR1 and DDR1)	85	Block Diagram of External Interrupt Circuit	185
DDR3		Interrupts Issued during External Interrupt Circuit	
Port 3 Registers (PDR3 and DDR3)	95	Operation	194
DDR4		Operations of the External Interrupt Circuit....	195
Port 4 Registers (PDR4 and DDR4)	102	Pins Related to External Interrupt Circuit	187
DDR5		Program Example of the External Interrupt Circuit	
Port 5 Registers (PDR5 and DDR5)	109	196
Dedicated Register		Register and Vector Table Related to Interrupt of	
Dedicated Register Configuration.....	28	External Interrupt Circuit	194
Dedicated Register Functions	28	Register Related to External Interrupt Circuit	
Descriptor Acquisition		190
Descriptor Acquisition.....	257	External Interrupt Control Register	
DMA Base Address Register		External Interrupt Control Register (EIE)	191
DMA Base Address Register (DBAR)	225	External Interrupt Flag Register	
DMA Base Address Register High		External Interrupt Flag Register (EIF)	193
DMA Base Address Register High (DBARH)	227	External Reset Pin	
DMA Base Address Register Low		Block Diagram of the External Reset Pin	48
DMA Base Address Register Low (DBAR)....	227	Functions of the External Reset Pin.....	48
DMA Transfer		F	
DMA Transfer to or from External FIFO	254	F ² MC-8L	
E		Overview of F ² MC-8L Instructions	313
EIE		FIFO	
External Interrupt Control Register (EIE).....	191	DMA Transfer to or from External FIFO	254
EIF		FPT-64P-M03	
External Interrupt Flag Register (EIF).....	193	Package Dimensions (FPT-64P-M03)	10
G		G	
General-Purpose Register		General-Purpose Register	
General-Purpose Register Configuration	34	General-Purpose Register Configuration	34
General-purpose Register		General-purpose Register	
General-purpose Register Area		General-purpose Register Area	
(address:0100 _H to 01FF _H)	24	(address:0100 _H to 01FF _H)	24

General-Purpose Registers	
Features of General-Purpose Registers	35
H	
Handling Devices	
Notes on Handling Devices	20
High-speed PWM Timer	
Operation of the High-speed PWM Timer	
Function	166
I	
I/O Circuit	
I/O Circuit Types	16
I/O Map	
I/O Map	308
I/O Port	
I/O Port Functions	76
I/O Ports	
Program Example of the I/O Ports	118
ILR	
Configuration of Interrupt Level Registers	
(ILR1,ILR2,and ILR3)	38
Instruction	
Instruction Map	329
Instruction Cycle	
Instruction Cycle (t_{inst})	58
Instructions	
Overview of F ² MC-8L Instructions	313
Intermediate Stop	
Operation for Standby Mode and Intermediate	
Stop	172
Interrupt	
Bits That Control Acceptance of an Interrupt	
.....	31
Interrupt During Parallel Port Operation	210
Interrupt during USB Function Operation	249
Interrupt Processing Time	43
Oscillation Stabilization Wait Time and Timebase	
Timer Interrupt	126
Processing During Interrupt Operation	40
Register and Vector Table for a Parallel Port	
Interrupt	210
Register and Vector Table Related to Interrupt of	
External Interrupt Circuit	194
Releasing Standby Mode with an Interrupt	70
Stack Area for Interrupt Processing	45
Stack Operation on Return from an Interrupt	
.....	44
Stack Operation When Interrupt Processing	
Starts	44
Interrupt Level	
Configuration of Interrupt Level Registers	
(ILR1,ILR2,and ILR3)	38

Interrupt Requests	
Interrupt Requests from Peripheral Functions	
.....	36
Interrupts	
Entering Standby Mode and Interrupts	70
Interrupts	161
Interrupts Issued during External Interrupt Circuit	
Operation	194
Interrupts Related to Operation of the Interval	
Timer Function	126
Receive Interrupts	284
Register and Vector Table Related to Interrupts	
.....	161
Register and Vector Table Related to Timebase	
Timer Interrupts	126
Registers and Vector Tables Related to USB	
Interrupts	250
Send Interrupts	284
Interval Timer	
Interrupts Related to Operation of the Interval	
Timer Function	126
Interval Timer Function	120
Interval Timer Function (Rectangular Output	
Function)	140
Operation of the Interval Timer Function	162
Operation of the Interval Timer Function	
(Timebase Timer)	127
L	
Level Detection	
Function of External Interrupt Circuit (Level	
Detection)	184
M	
Main Clock Mode	
Operation in Main Clock Mode	59
MB89580B/BW	
Full Block Diagram of MB89580B/BW Series ...	8
MB89580B/BW Series Features	2
MB89580B/BW Series Models	4
MB89580B/BW Series Pin Assignments	9
Memory Map	
Memory Map in PROM Mode	330
Memory Space	
Memory Space Configuration	22
MOD	
Mode Pins (MOD0,MOD1)	72
Mode Data	
Mode Data	72
Pin State After Mode Data Is Read	51
Mode Fetch	
Mode Fetch	50
Mode Operation	
CH12PWM Mode Operation	168

Mode Pins		
Mode Pins (MOD0,MOD1)	72	
Model Selection		
Precautions for Model Selection	6	
Multiple Interrupts		
Example of Multiple Interrupts	42	
Multiple Interrupts	42	
O		
Oscillation Stabilization Wait Time		
Oscillation Stabilization Wait Time	60, 71	
Oscillation Stabilization Wait Time and Timebase Timer Interrupt	126	
Oscillation Stabilization Wait Time Reset State	50	
P		
Package Dimensions		
Package Dimensions (FPT-64P-M03)	10	
Parallel Port		
Block Diagram of Parallel Port	201	
Interrupt During Parallel Port Operation	210	
Parallel Port Functions	200	
Parallel Port Input Operation	211	
Parallel Port Registers	204	
Register and Vector Table for a Parallel Port Interrupt	210	
parallel port		
parallel port pins	202	
Parallel Port Data Control Register		
Parallel Port Data Control Register (PDCR)	208	
Parallel Port Data Register		
Parallel Port Data Register (PDR)	205	
Parallel Port Status Register		
Parallel Port Status Register (PSR)	206	
PDCR		
Parallel Port Data Control Register (PDCR)	208	
PDR		
Parallel Port Data Register (PDR)	205	
PDR0		
Port 0 Registers (PDR0 and DDR0)	80	
PDR1		
Port 1 Registers (PDR1 and DDR1)	85	
PDR2		
Port 2 Register (PDR2)	90	
PDR3		
Port 3 Registers (PDR3 and DDR3)	95	
PDR4		
Port 4 Registers (PDR4 and DDR4)	102	
PDR5		
Port 5 Registers (PDR5 and DDR5)	109	
Peripheral Functions		
Interrupt Requests from Peripheral Functions	36	
Pin Assignments		
MB89580B/BW Series Pin Assignments	9	
Pin Functions		
Pin Functions	12	
Pin State		
Pin State After Mode Data Is Read	51	
Pin State During Reset	51	
Pin State for Each Operation	332	
Pins		
Pins Related to the 2-Channel 8-Bit PWM Timer	149	
PMDR		
USB Power Mode Register (PMDR)	222	
Port 0		
Functions of the Port 0 Registers	81	
Operation of Port 0	82	
Port 0 Configuration	79	
Port 0 Pins	79	
Port 0 Registers (PDR0 and DDR0)	80	
Port 1		
Functions of the Port 1 Registers	86	
Operation of Port 1	87	
Port 1 Configuration	84	
Port 1 Pins	84	
Port 1 Registers (PDR1 and DDR1)	85	
Port 2		
Function of Port 2 Register	91	
Operation of Port 2	92	
Port 2 Configuration	89	
Port 2 Pins	89	
Port 2 Register (PDR2)	90	
Port 3		
Functions of the Port 3 Registers	96	
Operation of Port 3	98	
Port 3 Configuration	93	
Port 3 Pins	93	
Port 3 Registers (PDR3 and DDR3)	95	
Port 4		
Functions of the Port 4 Registers	103	
Operation of Port 4	105	
Port 4 Configuration	100	
Port 4 Pins	100	
Port 4 Registers (PDR4 and DDR4)	102	
Port 5		
Block Diagram of Port 5	108	
Functions of the Port 5 Registers	110	
Operation of Port 5	112	
Port 5 Configuration	107	
Port 5 Pins	107	
Port 5 Registers (PDR5 and DDR5)	109	

Port 6	
Block Diagram of Port 6	115
Function of Port 6 Register	116
Operation of Port 6	117
Port 6 Configuration	114
Port 6 Pins	114
Port 6 Register	115
Prescaler Operation	
Prescaler Operation	170
Program Example	
Program Example of 8-bit Timer Mode	175
Program Example of PWM Timer Function	179
Program Example of the External Interrupt Circuit	196
Program Example of the I/O Ports	118
Program Example of the Timebase Timer	130
Program Example of the Watchdog Timer	138
programmer	
Programming to EPROM (When using the Ando Electric Co.,Ltd. programmer)	331
Programming	
Programming to EPROM (When using the Ando Electric Co.,Ltd. programmer)	331
Programming Yield	
Programming Yield	331
PROM	
Memory Map in PROM Mode	330
PROM Mode	
Memory Map in PROM Mode	330
Pull-up Option	
Pull-up Option	304
Pull-up Option Setting Registers (PURR 0, PURR 1, and PURR 2)	305
PURR	
Pull-up Option Setting Registers (PURR 0, PURR 1, and PURR 2)	305
PWM Compare Register	
PWM Compare Register 1,2 (COMR1, COMR2)	147
PWM Compare Register 1	
PWM Compare Register 1 (COMR1)	157
PWM Compare Register 2	
PWM Compare Register 2 (COMR2)	159
PWM Control Register	
PWM Control Register 1,2,3 (CNTR1, CNTR2, CNTR3)	147, 150
PWM Control Register 1	
PWM Control Register 1 (CNTR1)	151
PWM Control Register 2	
PWM Control Register 2 (CNTR2)	153
PWM Control Register 3	
PWM Control Register 3 (CNTR3)	155
PWM Timer	
8-Bit PWM Timer 1 (CH1) and 8-Bit PWM Timer 2 (CH2)	147
Block Diagram of Pins Related to the 2-Channel 8-Bit PWM Timer	149
Notes on Using the 2-Channel 8-Bit PWM Timer	174
Operation of the High-speed PWM Timer Function	166
Operation of the PWM Timer Function	164
Overview of the 2-Channel 8-Bit PWM Timer (PWM Timer Function)	143
Pins Related to the 2-Channel 8-Bit PWM Timer	149
Program Example of PWM Timer Function	179
PWM Timer Function	144
R	
RAM	
Effect of a Reset on the Contents of RAM	50
Read-modify-write	
Read-modify-write Operation	322
Receive Interrupts	
Receive Interrupts	284
Receive Operation	
Start Bit Detection in a Receive Operation	291
Recommended ROM Programmers	
ROM Programmer Adapters and Recommended ROM Programmers	330
Rectangular Output Function	
Interval Timer Function (Rectangular Output Function)	140
Register	
Port 2 Register (PDR2)	90
Port 6 Register	115
Register and Address of Vector Table Related to UART/SIO Interrupts	284
Register and Vector Table for a Parallel Port Interrupt	210
Register and Vector Table Related to Interrupt of External Interrupt Circuit	194
Register and Vector Table Related to Interrupts	161
Register and Vector Table Related to Timebase Timer Interrupts	126
Register Related to Clock Output Control Register (CKR)	302
Register Related to External Interrupt Circuit	190
Register Bank Pointer	
Configuration of Register Bank Pointer (RP)	33

Registers	
Configuration of Interrupt Level Registers	
(ILR1, ILR2, and ILR3)	38
Parallel Port Registers	204
Port 0 Registers (PDR0 and DDR0)	80
Port 1 Registers (PDR1 and DDR1)	85
Port 3 Registers (PDR3 and DDR3)	95
Port 4 Registers (PDR4 and DDR4)	102
Port 5 Registers (PDR5 and DDR5)	109
Registers and Vector Tables Related to USB	
Interrupts	250
USB Function Circuit Registers	218
Regulatory Recovery Time	
Regulatory Recovery Time	50
Regulatory Stabilization Time	
Regulatory Stabilization Time	50
Remote Wake-up	
Remote Wake-up from a Device	264
Reset	
Effect of a Reset on the Contents of RAM	50
Oscillation Stabilization Wait Time Reset State	
.....	50
Pin State During Reset	51
Reset Causes	46
USB Bus Reset	257
Reset Operation	
Overview of Reset Operation	49
ROM Programmer Adapters	
ROM Programmer Adapters and Recommended	
ROM Programmers	330
RP	
Configuration of Register Bank Pointer (RP)	
.....	33
S	
SCS	
Serial Clock Switch register (SCS)	278
UART/SIO Register (SCS, SMC1, SMC2, SSD,	
SIDR, SODR, SRC)	273
Send Interrupts	
Send Interrupts	284
Serial Clock Switch register	
Serial Clock Switch register (SCS)	278
Serial Input Data Register	
Serial Input Data Register (SIDR)	281
Serial Mode Control Register 1	
Serial Mode Control Register 1 (SMC1)	274
Serial Mode Control Register 2	
Serial Mode Control Register 2 (SMC2)	276
Serial Output Data Register	
Serial Output Data Register (SODR)	282
Serial Rate Control Register	
Serial Rate Control Register (SRC)	283
Serial Status and Data Register	
Serial Status and Data Register (SSD)	279
SIDR	
Serial Input Data Register (SIDR)	281
UART/SIO Register	
(SCS, SMC1, SMC2, SSD, SIDR, SODR,	
SRC)	273
Single-chip Mode	
Single-chip Mode	72
SIO	
Block Diagram of UART/SIO	269
Explanation of UART/SIO Operating Mode	
.....	293
Explanation of UART/SIO Operating Mode 0	
.....	286
Pins Related to UART/SIO	271
Register and Address of Vector Table Related to	
UART/SIO Interrupts	284
UART/SIO Functions	268
UART/SIO Operation	285
Sleep Mode	
Operation in Sleep Mode	64
SMC	
UART/SIO Register	
(SCS, SMC1, SMC2, SSD, SIDR, SODR,	
SRC)	273
SMC1	
Serial Mode Control Register 1 (SMC1)	274
SMC2	
Serial Mode Control Register 2 (SMC2)	276
SODR	
Serial Output Data Register (SODR)	282
UART/SIO Register	
(SCS, SMC1, SMC2, SSD, SIDR, SODR,	
SRC)	273
Special Instructions	
Special Instructions	319
SRC	
Serial Rate Control Register (SRC)	283
UART/SIO Register	
(SCS, SMC1, SMC2, SSD, SIDR, SODR,	
SRC)	273
SSD	
Serial Status and Data Register (SSD)	279
UART/SIO Register (SCS, SMC1, SMC2, SSD,	
SIDR, SODR, SRC)	273
Stack Area	
Stack Area for Interrupt Processing	45
Stack Operation	
Stack Operation on Return from an Interrupt	
.....	44
Stack Operation When Interrupt Processing	
Starts	44

Standby Control Register	
Standby Control Register (STBC).....	66
Standby Mode	
Entering Standby Mode and Interrupts	70
Note on Setting Standby Mode	70
Operation for Standby Mode and Intermediate	
Stop	172
Releasing Standby Mode with an Interrupt.....	70
Standby Mode.....	62
Standby Mode Operating State	63
Start Bit Detection	
Start Bit Detection in a Receive Operation	291
State Transition Chart	
State Transition Chart.....	68
STBC	
Standby Control Register (STBC).....	66
Stop Mode	
Operation in Stop Mode.....	65
Storage State	
Storage State of 16-Bit Data in RAM	26
Storage State of a 16-Bit Operand	26
Suspend Processing	
Suspend Processing.....	262
SYCC	
System Clock Control Register (SYCC)	57
System Clock Control Register	
System Clock Control Register (SYCC)	57
T	
TBTC	
Timebase Timer Control Register (TBTC)	124
TDCR	
Transfer Data Count Registers	
(TDCR0,TDCR1,TDCR2,TDCR3)	229
Timebase Timer	
Block Diagram of the Timebase Timer	122
Notes on Using the Timebase Timer	129
Operation of the Interval Timer Function	
(Timebase Timer)	127
Operation of Timebase Timer	127
Oscillation Stabilization Wait Time and Timebase	
Timer Interrupt	126
Program Example of the Timebase Timer	130
Register and Vector Table Related to Timebase	
Timer Interrupts	126
Timebase Timer Control Register	
Timebase Timer Control Register (TBTC)	124
Timer Mode	
Program Example of 8-bit Timer Mode	175
t_{inst}	
Instruction Cycle (t_{inst})	58
Transfer Data Count Registers	
Transfer Data Count Registers	
(TDCR0,TDCR1,TDCR2,TDCR3)	229
Transfer Instructions	
Transfer Instructions.....	323
U	
UART	
Block Diagram of UART/SIO	269
Explanation of UART/SIO Operating Mode	
.....	293
Explanation of UART/SIO Operating Mode 0	
.....	286
Pins Related to UART/SIO	271
Register and Address of Vector Table Related to	
UART/SIO Interrupts	284
UART/SIO Functions	268
UART/SIO Operation.....	285
UART/SIO	
Block Diagram of UART/SIO	269
Explanation of UART/SIO Operating Mode	
.....	293
Explanation of UART/SIO Operating Mode 0	
.....	286
Pins Related to UART/SIO	271
Register and Address of Vector Table Related to	
UART/SIO Interrupts	284
UART/SIO Functions	268
UART/SIO Operation.....	285
UART/SIO Register	
UART/SIO Register	
(SCS,SMC1,SMC2,SSD,SIDR,SODR,	
SRC).....	273
UCTR	
USB Control Register (UCTR).....	231
UFRMR	
USB Frame Status Register (UFRMR).....	240
UMDR	
USB Reset Mode Register (UMDR)	223
UMSKR	
USB Interrupt Mask Register (UMSKR)	237
USB	
Functions Supported by the USB	251
Operation of the USB Function	255
Registers and Vector Tables Related to USB	
Interrupts	250
USB Bus Reset	257
USB Control Register	
USB Control Register (UCTR).....	231
USB Frame Status Register	
USB Frame Status Register (UFRMR).....	240
USB Function	
Interrupt during USB Function Operation	249

USB Function Circuit	
Block Diagram of USB Function Circuit.....	216
Functions of the USB Function Circuit.....	214
USB Function Circuit Registers	218
USB Interrupt Mask Register	
USB Interrupt Mask Register (UMSKR)	237
USB Power Mode Register	
USB Power Mode Register (PMDR).....	222
USB Reset Mode Register	
USB Reset Mode Register (UMDR)	223
USB Status Register 1	
USB Status Register 1 (USTR1)	232
USB Status Register 2	
USB Status Register 2 (USTR2)	235
USTR1	
USB Status Register 1 (USTR1)	232
USTR2	
USB Status Register 2 (USTR2)	235
V	
Vector Table	
Register and Address of Vector Table Related to UART/SIO Interrupts	284
Register and Vector Table for a Parallel Port Interrupt	210

Register and Vector Table Related to Interrupt of External Interrupt Circuit	194
Register and Vector Table Related to Interrupts	161
Register and Vector Table Related to Timebase Timer Interrupts.....	126
Vector Table Area (address:FFC0 _H to FFFF _H)	24

Vector Tables	
Registers and Vector Tables Related to USB	
Interrupts	250

W

Watchdog Control Register	
Watchdog Control Register (WDTC).....	135
Watchdog Timer	
Block Diagram of the Watchdog Timer	133
Notes on Using the Watchdog Timer	137
Operations of the Watchdog Timer	136
Program Example of the Watchdog Timer	138
Watchdog Timer Function	132
WDTC	
Watchdog Control Register (WDTC).....	135

CM25-10142-3E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F²MC™-8L

8-BIT MICROCONTROLLER

MB89580B/BW Series

HARDWARE MANUAL

February 2005 the third edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Business Promotion Dept.
