OKI

MSM66577 Family

User's Manual

CMOS 16-bit microcontroller

Preliminary

FIRST EDITION

ISSUE DATE: Dec. 2000

PEUL66577-01

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans. Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.

9. MS-DOS is a registered trademark of Microsoft Corporation.

Printed in Japan

Preface

This user's manual describes the hardware of Oki-original CMOS 16-bit microcontrollers MSM66577 family. In addition to this manual, Oki also provides the following manuals which should be read with regard to the MSM66577 family.

nX-8/500S Core Instruction Manual

- nX-8/500S core instruction set
- Addressing modes

CC665S User's Manual

- Optimized compiler CC665S operation
- C-language specifications in CC665S

CL665S User's Manual

- Compiler loader CL665S operation

RTL665S Run Time Library Reference

- C run time library explanation

MAC66K Assembler Package User’s Manual

- Package overview
- RAS66K (relocatable assembler) operation
- RAS66K assembly language explanation
- RL66K (linker) operation
- LIB66K (librarian) operation
- OH66K (object converter) operation

Macroprocessor MP User’s Manual

- MP operation
- Macro language

Ultra-66K/E502 User’s Manual

- Ultra-66K (Emulator) explanation
- PathFinder-66K (Debugger) explanation

PW66K Flash Writer System User’s Manual

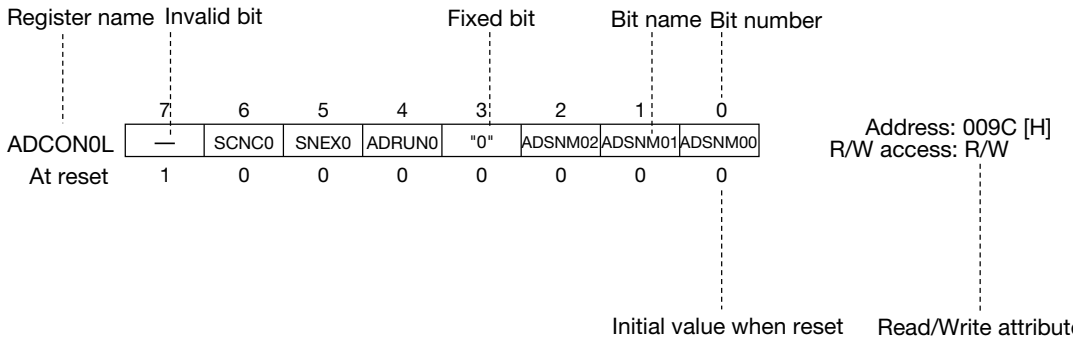
- PW66K Flash Writer System operation

This document is subject to change without notice.

Notation

Classification	Notation	Description
■ Numeric value	xxH	Represents a hexadecimal number
	xxb	Represents a binary number
■ Unit	Word, W	1 word = 16 bits
	byte, B	1 byte = 2 nibbles = 8 bits
	nibble, N	1 nibble = 4 bits
	mega-, M	10^6
	kilo-, K	$2^{10} = 1024$
	kilo-, k	$10^3 = 1000$
	milli-, m	10^{-3}
	micro-, μ	10^{-6}
	nano-, n	10^{-9}
	second, s	second
	KB	1KB = 1 kilobyte = 1024 bytes
	MB	1MB = 1 megabyte = 2^{20} bytes = 1,048,576 bytes
■ Terminology	"H" level	The signal level of the high side of the voltage; indicates the voltage level of V_{IH} and V_{OH} described in the electrical characteristics.
	"L" level	The signal level of the low side of the voltage; indicates voltage level of V_{IL} and V_{OL} described in the electrical characteristics.
	Opcode trap	Operation code trap. Occurs when an empty area that has not been assigned an instruction is fetched, or when an instruction code combination that does not contain an instruction is addressed.

■ Register description



Invalid bit : Indicates that the bit does not exist. Writing into this bit is invalid.
Fixed bit : When writing, always write the specified value. If read, the specified value will be read. Values of fixed bits are specified as "0" or "1."
Read/write attribute : R indicates that reading is possible and W indicates that writing is possible.

Contents

Chapter 1 Overview

1.1 Overview 1-1
1.2 Features 1-1
1.3 Block Diagram 1-4
1.4 Pin Configuration 1-5
1.5 Pin Descriptions 1-6
1.5.1 Description of Each Pin 1-6
1.5.2 Pin Configuration 1-9
1.5.3 Connections for Unused Pins 1-11
1.6 Basic Operational Timing 1-12

Chapter 2 CPU Architecture

2.1 Overview 2-1
2.2 Memory Space 2-1
2.2.1 Memory Space Expansion 2-1
2.2.2 Program Memory Space 2-3
(1) Accessing program memory space 2-5
(2) Vector table area 2-5
(3) VCAL table area 2-8
(4) ACAL area 2-9
2.2.3 Data Memory Space 2-10
(1) Special function register (SFR) area 2-12
(2) Reserved area 2-12
(3) Internal RAM area 2-12
(4) Fixed page (FIX) area 2-12
(5) Local register setting area 2-14
(6) External data memory area 2-14
(7) Common area 2-15
2.2.4 Data Memory Access 2-15
(1) Byte operations 2-15
(2) Word operations 2-16
2.3 Registers 2-17
2.3.1 Arithmetic Register (ACC) 2-17
2.3.2 Control Registers 2-18
(1) Program status word (PSW) 2-18
(2) Program counter (PC) 2-22

(3) Local register base (LRB) 2-22
(4) System stack pointer (SSP) 2-23
2.3.3 Pointing Register (PR) 2-24
2.3.4 Local Registers (R0 to R7, ER0 to ER3) 2-25
2.3.5 Segment Registers 2-26
(1) Code segment register (CSR) 2-26
(2) Table segment register (TSR) 2-26
(3) Data segment register (DSR) 2-27
2.4 Addressing Modes 2-27
2.4.1 RAM Addressing 2-27
(1) Register addressing 2-28
(2) Page addressing 2-30
(3) Direct data addressing 2-33
(4) Pointing register indirect addressing 2-34
(5) Special bit area addressing 2-41
2.4.2 ROM Addressing 2-43
(1) Immediate addressing 2-43
(2) Table data addressing 2-43
(3) Program code addressing 2-45
(4) ROM window addressing 2-46

Chapter 3 CPU Control Functions

3.1 Overview 3-1
3.2 Standby Functions 3-1
3.2.1 Standby Function Registers 3-3
3.2.2 Description of Standby Function Registers 3-3
(1) Stop code acceptor (STPACP) 3-3
(2) Standby control register (SBYCON) 3-4
3.2.3 Examples of Standby Function Register Settings 3-6
• HALT mode setting 3-6
• HOLD mode setting 3-6
• STOP mode setting 3-6
3.2.4 Operation of Each Standby Mode 3-6
(1) HALT mode 3-6
(2) HOLD mode 3-7
(3) STOP mode 3-8
3.3 Reset Function 3-10

Chapter 4 Memory Control Functions

4.1 Overview 4-1
4.2 Memory Control Function Registers 4-1
4.3 ROM Window Function 4-2
4.4 READY Function 4-4
4.4.1 ROM Ready Control Register (ROMRDY) 4-4
4.4.2 RAM Ready Control Register (RAMRDY) 4-5
4.5 WAIT Function 4-7

Chapter 5 Port Functions

5.1 Overview 5-1
5.2 Hardware Configuration of Each Port 5-3
5.2.1 Type A (P0) 5-3
5.2.2 Type B (P1, P2, P3_0, P3_1, P4) 5-4
5.2.3 Type C (P3_2, P3_3) 5-5
5.2.4 Type D
(P5, P6, P7, P8, P9, P10, P11, P14_0 to P14_2, P15) 5-6
5.2.5 Type E (P14_6, P14_7) 5-7
5.2.6 Type F (P12) 5-7
5.3 Port Registers 5-8
5.3.1 Port Data Registers (Pn:n = 0 to 12, 14, 15) 5-10
5.3.2 Port Mode Registers (PnIO:n = 0 to 11, 14, 15) 5-10
5.3.3 Port Secondary Function Control Registers (PnSF:n = 0 to 11, 14, 15) ... 5-11
5.4 Port 0 (P0) 5-12
5.5 Port 1 (P1) 5-14
5.6 Port 2 (P2) 5-16
5.7 Port 3 (P3) 5-18
5.8 Port 4 (P4) 5-20
5.9 Port 5 (P5) 5-22
5.10 Port 6 (P6) 5-24
5.11 Port 7 (P7) 5-26
5.12 Port 8 (P8) 5-28
5.13 Port 9 (P9) 5-30
5.14 Port 10 (P10) 5-32
5.15 Port 11 (P11) 5-34
5.16 Port 12 (P12) 5-36
5.17 Port 14 (P14) 5-37
5.18 Port 15 (P15) 5-39

Chapter 6 Clock Oscillation Circuit

6.1 Overview 6-1
6.2 Clock Oscillation Circuit Configuration 6-1
6.3 Clock Oscillation Circuit Registers 6-2
6.4 OSC Oscillation Circuit 6-2
6.5 XT Oscillation Circuit 6-4

Chapter 7 Time Base Counter (TBC)

7.1 Overview 7-1
7.2 Time Base Counter (TBC) Configuration 7-1
7.3 Time Base Counter Registers 7-2
7.4 1/n Counter 7-2
7.4.1 Description of 1/n Counter Registers 7-2
(1) TBC clock dividing counter (TBCKDV upper 8 bits) 7-2
(2) TBC clock divider register (TBCKDVR) 7-3
7.4.2 Example of 1/n Counter-related Register Settings 7-4
7.5 Time Base Counter (TBC) Operation 7-4

Chapter 8 General-Purpose 8/16 Bit Timers

8.1 Overview 8-1
8.2 General-purpose 8-bit/16-bit Timer Configurations 8-1
8.3 General-purpose 8-bit/16-bit Timer Registers 8-2
8.4 Timer 0 8-3
8.4.1 Timer 0 Configuration 8-3
8.4.2 Description of Timer 0 Registers 8-4
(1) General-purpose 16-bit timer 0 counter (TM0C) 8-4
(2) General-purpose 16-bit timer 0 register (TM0R) 8-4
(3) General-purpose 16-bit timer 0 control register (TM0CON) 8-4
8.4.3 Example of Timer 0-related Register Settings 8-6
(1) Port 5 mode register (P5IO) 8-6
(2) Port 5 secondary function control register (P5SF) 8-6
(3) General-purpose 16-bit timer 0 counter (TM0C) 8-6
(4) General-purpose 16-bit timer 0 register (TM0R) 8-6
(5) General-purpose 16-bit timer 0 control register (TM0CON) 8-6
8.4.4 Timer 0 Operation 8-7
8.4.5 Timer 0 Interrupt 8-8
8.5 Timers 1 and 2 8-9
8.5.1 Timers 1 and 2 Configurations 8-9

8.5.2 Description of Timer 1 and 2 Registers 8-10
(1) General-purpose 8-bit timer 1 and 2 counters (TM1C, TM2C) 8-10
(2) General-purpose 8-bit timer 1 and 2 registers (TM1R, TM2R) 8-10
(3) General-purpose 8-bit timer 1 control register (TM1CON) 8-10
(4) General-purpose 8-bit timer 2 control register (TM2CON) 8-11
8.5.3 Example of Timer 1- and 2-related Register Settings 8-13
• 8-bit auto-reload timer mode (Timer 1) 8-13
(1) Port 6 mode register (P6IO) 8-13
(2) Port 6 secondary function control register (P6SF) 8-13
(3) General-purpose 8-bit timer 1 counter (TM1C) 8-13
(4) General-purpose 8-bit timer 1 register (TM1R) 8-13
(5) General-purpose 8-bit timer 1 control register (TM1CON) 8-13
• 8-bit auto-reload timer mode (Timer 2) 8-13
(1) Port 6 mode register (P6IO) 8-13
(2) Port 6 secondary function control register (P6SF) 8-13
(3) General-purpose 8-bit timer 2 counter (TM2C) 8-13
(4) General-purpose 8-bit timer 2 register (TM2R) 8-14
(5) General-purpose 8-bit timer 2 control register (TM2CON) 8-14
• 16-bit auto-reload timer mode 8-14
(1) Port 6 mode register (P6IO) 8-14
(2) Port 6 secondary function control register (P6SF) 8-14
(3) General-purpose 16-bit timer 12 counter (TM12C) 8-14
(4) General-purpose 16-bit timer 12 register (TM12R) 8-14
(5) General-purpose 8-bit timer 1 control register (TM1CON) 8-14
(6) General-purpose 8-bit timer 2 control register (TM2CON) 8-14
• PWM mode 8-15
(1) Port 6 mode register (P6IO) 8-15
(2) Port 6 secondary function control register (P6SF) 8-15
(3) General-purpose 16-bit timer 12 counter (TM12C) 8-15
(4) General-purpose 16-bit timer 12 register (TM12R) 8-15
(5) General-purpose 8-bit timer 1 control register (TM1CON) 8-15
(6) General-purpose 8-bit timer 2 control register (TM2CON) 8-15
8.5.4 Timer 1 and 2 Operation 8-16
• 8-bit auto-reload timer mode 8-16
• 16-bit auto-reload timer mode 8-17
• PWM mode 8-18
8.5.5 Timer 1 and 2 Interrupts 8-19
• Timer 1 interrupt 8-19
• Timer 2 interrupt 8-20

8.6 Timer 3 8-21
8.6.1 Timer 3 Configuration 8-21
8.6.2 Description of Timer 3 Registers 8-22
(1) General-purpose 8-bit timer 3 counter (TM3C) 8-22
(2) General-purpose 8-bit timer 3 register (TM3R) 8-22
(3) General-purpose 8-bit timer 3 control register (TM3CON) 8-22
8.6.3 Example of Timer 3-related Register Settings 8-24
(1) General-purpose 8-bit timer 3 counter (TM3C) 8-24
(2) General-purpose 8-bit timer 3 register (TM3R) 8-24
(3) General-purpose 8-bit timer 3 control register (TM3CON) 8-24
8.6.4 Timer 3 Operation 8-25
8.6.5 Timer 3 Interrupt 8-26
8.7 Timer 4 8-27
8.7.1 Timer 4 Configuration 8-27
8.7.2 Description of Timer 4 Registers 8-28
(1) General-purpose 8-bit timer 4 counter (TM4C) 8-28
(2) General-purpose 8-bit timer 4 register (TM4R) 8-28
(3) General-purpose 8-bit timer 4 control register (TM4CON) 8-28
8.7.3 Example of Timer 4-related Register Settings 8-30
(1) Port 8 mode register (P8IO) 8-30
(2) Port 8 secondary function control register (P8SF) 8-30
(3) General-purpose 8-bit timer 4 counter (TM4C) 8-30
(4) General-purpose 8-bit timer 4 register (TM4R) 8-30
(5) General-purpose 8-bit timer 4 control register (TM4CON) 8-30
8.7.4 Timer 4 Operation 8-31
8.7.5 Timer 4 Interrupt 8-32
8.8 Timer 5 8-33
8.8.1 Timer 5 Configuration 8-33
8.8.2 Description of Timer 5 Registers 8-34
(1) General-purpose 8-bit timer 5 counter (TM5C) 8-34
(2) General-purpose 8-bit timer 5 register (TM5R) 8-34
(3) General-purpose 8-bit timer 5 control register (TM5CON) 8-34
8.8.3 Example of Timer 5-related Register Settings 8-36
(1) General-purpose 8-bit timer 5 counter (TM5C) 8-36
(2) General-purpose 8-bit timer 5 register (TM5R) 8-36
(3) General-purpose 8-bit timer 5 control register (TM5CON) 8-36
8.8.4 Timer 5 Operation 8-37
8.8.5 Timer 5 Interrupt 8-38
8.9 Timer 6 8-39
8.9.1 Timer 6 Configuration 8-39

8.9.2 Description of Timer 6 Registers 8-40
(1) General-purpose 8-bit timer 6 counter (TM6C) 8-40
(2) General-purpose 8-bit timer 6 register (TM6R) 8-40
(3) General-purpose 8-bit timer 6 control register (TM6CON) 8-41
8.9.3 Example of Timer 6-related Register Settings 8-43
• Auto-reload timer mode settings 8-43
(1) General-purpose 8-bit timer 6 counter (TM6C) 8-43
(2) General-purpose 8-bit timer 6 register (TM6R) 8-43
(3) General-purpose 8-bit timer 6 control register (TM6CON) 8-43
• Watchdog timer (WDT) mode settings 8-43
(1) General-purpose 8-bit timer 6 register (TM6R) 8-43
(2) General-purpose 8-bit timer 6 control register (TM6CON) 8-43
(3) General-purpose 8-bit timer 6 counter (TM6C) 8-43
8.9.4 Timer 6 Operation 8-44
• Auto-reload timer mode 8-44
• Watchdog timer (WDT) mode 8-44
8.9.5 Timer 6 Interrupt (During Auto-Reload Timer Mode) 8-47
8.10 Timer 9 8-48
8.10.1 Timer 9 Configuration 8-48
8.10.2 Description of Timer 9 Registers 8-49
(1) General-purpose 8-bit timer 9 counter (TM9C) 8-49
(2) General-purpose 8-bit timer 9 register (TM9R) 8-49
(3) General-purpose 8-bit timer 9 control register (TM9CON) 8-49
8.10.3 Example of Timer 9-related Register Settings 8-51
(1) General-purpose 8-bit timer 9 counter (TM9C) 8-51
(2) General-purpose 8-bit timer 9 register (TM9R) 8-51
(3) General-purpose 8-bit timer 9 control register (TM9CON) 8-51
8.10.4 Timer 9 Operation 8-52
8.10.5 Timer 9 Interrupt 8-53

Chapter 9 Capture/Compare Timer

9.1 Overview 9-1
9.2 Capture/Compare Timer Configuration 9-1
9.3 Capture/Compare Timer Registers 9-2
9.4 16-Bit Free Running Counter (FRC) 9-3
9.4.1 16-Bit Free Running Counter Configuration 9-3
9.4.2 Description of 16-bit Free Running Counter Register 9-3
(1) 16-bit free running counter (FRC) 9-3
(2) Free running counter control register (FRCON) 9-4

9.5 Capture/Compare Out Modules 9-5
9.5.1 Capture/Compare Out Module Configuration 9-5
9.5.2 Description of Capture/Compare Out Module Registers 9-6
(1) Capture/compare registers (CPCMR0, CPCMR1) 9-6
(2) Capture/compare control registers (CPCMCON0, CPCMCON1) 9-6
(3) Capture control register (CAPCON) 9-7
(4) Capture/compare buffer registers (CPCMBFR0, CPCMBFR1) 9-8
9.6 Example of Capture/Compare Timer-related Register Settings 9-8
9.6.1 Capture Mode Settings 9-8
(1) Port 5 mode register (P5IO) 9-8
(2) Port 5 secondary function control register (P5SF) 9-8
(3) Capture control register (CAPCON) 9-8
(4) Free running counter (FRC) 9-8
(5) Free running counter control register (FRCON) 9-8
9.6.2 Compare Out Mode Settings 9-9
(1) Port 5 mode register (P5IO) 9-9
(2) Port 5 secondary function control register (P5SF) 9-9
(3) Capture/compare control registers (CPCMCON0, CPCMCON1) 9-9
(4) Free running counter (FRC) 9-9
(5) Capture/compare registers (CPCMR0, CPCMR1) 9-9
(6) Capture/compare buffer registers (CPCMBFR0, CPCMBFR1) 9-9
(7) Free running counter control register (FRCON) 9-9
9.7 Capture/Compare Timer Operation 9-10
9.7.1 Capture Mode Operation 9-10
9.7.2 Compare Out Mode Operation 9-11
9.8 Example Timings for Changing the Output Level of Compare Out 9-12
9.9 Capture/Compare Timer Interrupt 9-14

Chapter 10 Real-Time Counter (RTC)

10.1 Overview 10-1
10.2 Real-Time Counter Configuration 10-1
10.3 Real-Time Counter Control Register (RTCCON) 10-2
10.4 Example of Real-Time Counter Register Settings 10-3
10.5 Real-Time Counter Operation 10-3
10.6 Real-Time Counter Interrupt 10-4

Chapter 11 PWM Function

11.1 Overview 11-1
11.2 PWM Configuration 11-1

11.3 PWM Register 11-2
11.3.1 Description of PWM Registers 11-3
(1) PWM counters (PWC0, PWC1) 11-3
(2) PWM cycle registers (PWCY0, PWCY1) 11-3
(3) PWM registers (PWR0 to PWR3) 11-4
(4) PWM control register 0 (PWCON0) 11-4
(5) PWM control register 1 (PWCON1) 11-6
11.3.2 Example of PWM-related Register Settings 11-7
• 8-bit PWM settings 11-7
(1) Port 7 mode register (P7IO) 11-7
(2) Port 8 mode register (P8IO) 11-7
(3) Port 7 secondary function control register (P7SF) 11-7
(4) Port 8 secondary function control register (P8SF) 11-7
(5) PWM counters (PWC0, PWC1) 11-7
(6) PWM cycle registers (PWCY0, PWCY1) 11-7
(7) PWM registers (PWR0 to PWR3) 11-7
(8) PWM control register 0 (PWCON0) 11-8
• 16-bit PWM settings 11-8
(1) Port 7 mode register (P7IO) 11-8
(2) Port 8 mode register (P8IO) 11-8
(3) Port 7 secondary function control register (P7SF) 11-8
(4) Port 8 secondary function control register (P8SF) 11-8
(5) PWM counters (PWC0, PWC1) 11-8
(6) PWM cycle register (PWCY) 11-8
(7) PWM registers (PWR01, PWR23) 11-8
(8) PWM control register 0 (PWCON0) 11-9
(9) PWM control register 1 (PWCON1) 11-9
11.4 PWM Operation 11-9
11.4.1 PWM Operation During 8-bit Mode 11-9
11.4.2 PWM Operation During 16-bit Mode 11-10
11.4.3 PWM Operation During High-Speed Mode 11-12
11.5 PWM Interrupts 11-14

Chapter 12 Serial Port Functions

12.1 Overview 12-1
12.2 Serial Port Configuration 12-1
12.3 Serial Port Registers 12-2
12.4 SIO1 12-3
12.4.1 SIO1 Configuration 12-3

12.4.2 Description of SIO1 Registers 12-4
(1) SIO1 transmit control register (ST1CON) 12-4
(2) SIO1 receive control register (SR1CON) 12-6
(3) SIO1 status register (S1STAT) 12-8
(4) SIO1 transmit-receive buffer register (S1BUF) 12-10
(5) SIO1 transmit shift register, receive shift register 12-10
12.4.3 Example of SIO1-related Register Settings 12-11
12.4.3.1 UART Mode Settings 12-11
• Transmit settings 12-11
(1) Port 8 mode register (P8IO) 12-11
(2) Port 8 secondary function control register (P8SF) 12-11
(3) SIO1 transmit control register (ST1CON) 12-11
(4) SIO1 receive control register (SR1CON) 12-11
(5) SIO1 transmit-receive buffer register (S1BUF) 12-11
• Receive settings 12-11
(1) Port 8 mode register (P8IO) 12-11
(2) Port 8 secondary function control register (P8SF) 12-11
(3) SIO1 receive control register (SR1CON) 12-12
12.4.3.2 Synchronous Mode Settings 12-12
• Transmit settings 12-12
(1) Port 8 mode register (P8IO) 12-12
(2) Port 8 secondary function control register (P8SF) 12-12
(3) SIO1 transmit control register (ST1CON) 12-12
(4) SIO1 transmit-receive buffer register (S1BUF) 12-12
• Receive settings 12-13
(1) Port 8 mode register (P8IO) 12-13
(2) Port 8 secondary function control register (P8SF) 12-13
(3) SIO1 receive control register (SR1CON) 12-13
12.4.3.3 Baud Rate Generator (Timer 4) Settings 12-13
(1) General-purpose 8-bit timer 4 counter (TM4C) 12-13
(2) General-purpose 8-bit timer 4 control register (TM4CON) 12-13
12.4.4 SIO1 Interrupt 12-14
12.5 SIO6 12-15
12.5.1 SIO6 Configuration 12-15
12.5.2 Description of SIO6 Registers 12-16
(1) SIO6 transmit control register (ST6CON) 12-16
(2) SIO6 receive control register (SR6CON) 12-18
(3) SIO6 status register (S6STAT) 12-20
(4) SIO6 transmit-receive buffer register (S6BUF) 12-22
(5) SIO6 transmit shift register, receive shift register 12-22

12.5.3 Example of SIO6-related Register Settings 12-23
12.5.3.1 UART Mode Settings 12-23
• Transmit settings 12-23
(1) Port 15 mode register (P15IO) 12-23
(2) Port 15 secondary function control register (P15SF) 12-23
(3) SIO6 transmit control register (ST6CON) 12-23
(4) SIO6 receive control register (SR6CON) 12-23
(5) SIO6 transmit-receive buffer register (S6BUF) 12-23
• Receive settings 12-23
(1) Port 15 mode register (P15IO) 12-23
(2) Port 15 secondary function control register (P15SF) 12-23
(3) SIO6 receive control register (SR6CON) 12-24
12.5.3.2 Synchronous Mode Settings 12-24
• Transmit settings 12-24
(1) Port 15 mode register (P15IO) 12-24
(2) Port 15 secondary function control register (P15SF) 12-24
(3) SIO6 transmit control register (ST6CON) 12-24
(4) SIO6 transmit-receive buffer register (S6BUF) 12-24
• Receive settings 12-25
(1) Port 15 mode register (P15IO) 12-25
(2) Port 15 secondary function control register (P15SF) 12-25
(3) SIO6 receive control register (SR6CON) 12-25
12.5.3.3 Baud Rate Generator (Timer 3) Settings 12-25
(1) General-purpose 8-bit timer 3 counter (TM3C) 12-25
(2) General-purpose 8-bit timer 3 control register (TM3CON) 12-25
12.5.4 SIO6 Interrupt 12-26
12.6 SIO1, SIO6 Operation 12-27
12.6.1 Transmit Operation 12-27
• UART mode 12-27
• Synchronous mode 12-28
12.6.2 Receive Operation 12-35
• UART mode 12-35
• Synchronous mode 12-36
12.7 SIO4 12-40
12.7.1 SIO4 Configuration 12-40
12.7.2 Description of SIO4 Registers 12-41
(1) SIO4 control register (SIO4CON) 12-41
(2) FIFO control register (FIFOCON) 12-43
(3) Serial input FIFO data register (SIN4) 12-45
(4) Serial output FIFO data register (SOUT4) 12-45

12.7.3 Example of SIO4-related Register Settings 12-46
• Master mode settings 12-46
(1) Port 10 mode register (P10IO) 12-46
(2) Port 10 secondary function control register (P10SF) 12-46
(3) Serial output FIFO data register (SOUT4) 12-46
(4) SIO4 control register (SIO4CON) 12-46
• Slave mode settings 12-47
(1) Port 10 mode register (P10IO) 12-47
(2) Port 10 secondary function control register (P10SF) 12-47
(3) Sserial output FIFO data register (SOUT4) 12-47
(4) SIO4 control register (SIO4CON) 12-47
12.7.4 SIO4 Interrupt 12-48
12.7.5 SIO4 Operation 12-49
12.8 SIO5 12-51
12.8.1 SIO5 Configuration 12-51
12.8.2 Description of SIO5 Registers 12-52
(1) SIO5 control register (SIO5CON) 12-52
(2) Serial input FIFO data register (SIN5) 12-54
(3) Serial output FIFO data register (SOUT5) 12-54
(4) FIFO mode control register (FIFOMOD) 12-54
12.8.3 Example of SIO5-related Register Settings 12-57
• Master mode settings 12-57
(1) Port 14 mode register (P14IO) 12-57
(2) Port 14 secondary function control register (P14SF) 12-57
(3) Serial output FIFO data register (SOUT5) 12-57
(4) SIO5 control register (SIO5CON) 12-57
• Slave mode settings 12-58
(1) Port 14 mode register (P14IO) 12-58
(2) Port 14 secondary function control register (P14SF) 12-58
(3) Serial output FIFO data register (SOUT5) 12-58
(4) SIO5 control register (SIO5CON) 12-58
12.8.4 SIO5 Interrupt 12-59
12.8.5 SIO5 Operation 12-60

Chapter 13 A/D Converter Functions

13.1 Overview 13-1
13.2 A/D Converter Configuration 13-1
13.3 A/D Converter Registers 13-2
13.3.1 Description of A/D Converter Registers 13-3
(1) A/D control register 0L (ADCON0L) 13-3

(2) A/D control register 0H (ADCON0H) 13-5
(3) A/D interrupt control register (ADINT0) 13-7
(4) A/D result registers (ADR00 to ADR07) 13-8
13.3.2 Example of A/D Converter-related Register Settings 13-9
• Scan mode setting 13-9
(1) A/D control register 0H (ADCON0H) 13-9
(2) A/D interrupt control register (ADINT0) 13-9
(3) A/D control register 0L (ADCON0L) 13-9
• Select mode setting 13-9
(1) A/D interrupt control register (ADINT0) 13-9
(2) A/D control register 0H (ADCON0H) 13-9
13.4 A/D Converter Operation 13-10
13.5 Notes Regarding Usage of A/D Converter 13-11
13.5.1 Considerations When Setting the Conversion Time 13-11
13.5.2 Noise-Suppression Measures 13-13
13.6 A/D Converter Interrupt 13-14

Chapter 14 D/A Converter Functions

14.1 Overview 14-1
14.2 D/A Converter Configuration 14-1
14.3 D/A Converter Registers 14-2
14.3.1 Description of D/A Converter Registers 14-2
(1) DA registers (DAR0, DAR1) 14-2
(2) DA control register (DACON) 14-2
14.3.2 Example of D/A Converter-related Register Settings 14-3
(1) Port 14 mode register (P14IO) 14-3
(2) Port 14 secondary function control register (P14SF) 14-3
(3) DA registers (DAR0, DAR1) 14-3
(4) DA control register (DACON) 14-3
14.3.3 D/A Converter Operation 14-4

Chapter 15 Peripheral Functions

15.1 Overview 15-1
15.2 Description of Each Peripheral Function 15-1
15.2.1 Clock Out Function 15-1
15.2.2 External XTCLK Input Control Function 15-1
15.2.3 HOLD Input Control Function 15-1
15.2.4 WAIT Input Control Function 15-1
15.3 Peripheral Control Register (PRPHCON) 15-2

Chapter 16 External Interrupt Functions

16.1 Overview 16-1
16.2 External Interrupt Registers 16-1
16.2.1 Description of External Interrupt Registers 16-2
(1) External interrupt control register 0 (EXI0CON) 16-2
(2) External interrupt control register 1 (EXI1CON) 16-3
(3) External interrupt control register 2 (EXI2CON) 16-4
16.2.2 Example of External Interrupt-related Register Settings 16-5
(1) Port 6 mode register (P6IO) 16-5
(2) Port 9 mode register (P9IO) 16-5
(3) Port 6 secondary function control register (P6SF) 16-5
(4) Port 9 secondary function control register (P9SF) 16-5
(5) External interrupt control register 0 (EXI0CON) 16-5
(6) External interrupt control register 1 (EXI1CON) 16-5
(7) External interrupt control register 2 (EXI2CON) 16-5
16.3 EXINT0 to EXINT7 Interrupts 16-6

Chapter 17 Interrupt Processing Functions

17.1 Overview 17-1
17.2 Interrupt Function Registers 17-2
17.3 Description of Interrupt Processing 17-3
17.3.1 Non-Maskable Interrupt (NMI) 17-3
17.3.2 Maskable Interrupts 17-5
(1) Interrupt request registers (IRQ0 to IRQ4) 17-5
(2) Interrupt enable registers (IE0 to IE4) 17-5
(3) Master interrupt enable flag (MIE) 17-5
(4) Master interrupt priority flag (MIPF) 17-5
(5) Interrupt priority control registers (IP0 to IP9) 17-6
17.3.3 Priority Control of Maskable Interrupts 17-10
(1) Basic interrupt control 17-10
(2) Multiple interrupt control 17-10
17.4 IRQ, IE and IP Register Configurations for Each Interrupt 17-12
17.4.1 Interrupt Request Registers (IRQ0 to IRQ4) 17-12
(1) Interrupt request register 0 (IRQ0) 17-12
(2) Interrupt request register 1 (IRQ1) 17-13
(3) Interrupt request register 2 (IRQ2) 17-14
(4) Interrupt request register 3 (IRQ3) 17-15
(5) Interrupt request register 4 (IRQ4) 17-16
17.4.2 Interrupt Enable Registers (IE0 to IE4) 17-17
(1) Interrupt enable register 0 (IE0) 17-17

(2) Interrupt enable register 1 (IE1) 17-18
(3) Interrupt enable register 2 (IE2) 17-19
(4) Interrupt enable register 3 (IE3) 17-20
(5) Interrupt enable register 4 (IE4) 17-21
17.4.3 Interrupt Priority Control Registers (IP0 to IP9) 17-22
(1) Interrupt priority control register 0 (IP0) 17-22
(2) Interrupt priority control register 1 (IP1) 17-23
(3) Interrupt priority control register 2 (IP2) 17-24
(4) Interrupt priority control register 3 (IP3) 17-25
(5) Interrupt priority control register 4 (IP4) 17-26
(6) Interrupt priority control register 5 (IP5) 17-27
(7) Interrupt priority control register 6 (IP6) 17-28
(8) Interrupt priority control register 7 (IP7) 17-29
(9) Interrupt priority control register 8 (IP8) 17-30
(10) Interrupt priority control register 9 (IP9) 17-31

Chapter 18 Bus Port Functions

18.1 Overview 18-1
18.2 Port Operation 18-1
18.2.1 Port Operation When Accessing Program Memory 18-1
18.2.2 Port Operation When Accessing Data Memory 18-4
18.3 External Memory Access 18-6
18.3.1 External Program Memory Access 18-6
18.3.2 External Data Memory Access 18-8
18.4 External Memory Access Timing 18-10
18.4.1 External Program Memory Access Timing 18-10
18.4.2 External Data Memory Access Timing 18-12
18.5 Notes Regarding Usage of Bus Port Function 18-16
18.5.1 Dummy Read Strobe Output 18-16
18.5.2 External Bus Access Timing 18-18

Chapter 19 Flash Memory

19.1 Overview 19-1
19.2 Features 19-1
19.3 Programming Modes 19-2
19.4 Parallel Mode 19-4
19.4.1 Overview of the Parallel Mode 19-4
19.4.2 PROM Writer Setting 19-4
19.4.3 Flash Memory Programming Conversion Adapter 19-4

19.5 Serial Mode 19-5
19.5.1 Overview of the Serial Mode 19-5
19.5.2 Serial Mode Settings 19-5
(1) Pins used in serial mode 19-5
(2) Serial mode connection circuit 19-6
(3) Serial mode programming method 19-8
(4) Setting of security function 19-8
(5) Notes on use of serial mode 19-8
19.6 User Mode 19-9
19.6.1 Overview of the User Mode 19-9
19.6.2 User Mode Programming Registers 19-10
19.6.3 Description of User Mode Registers 19-11
(1) Flash memory address register (FLAADRS) 19-11
(2) Flash memory acceptor (FLAACP) 19-11
(3) Flash memory control register (FLACON) 19-12
19.6.4 User Mode Programming Example 19-15
(1) User mode programming flowchart example 19-15
(2) User mode programming program example 19-16
19.6.5 Notes on Use of User Mode 19-16
19.7 Notes on Program 19-17
(1) Programming of flash memory immediately after power-on 19-17
(2) Note on STOP mode release 19-17
(3) Supply voltage sense reset function 19-17

Chapter 20 Electrical Characteristics

20.1 Absolute Maximum Ratings 20-1
20.2 Recommended Operating Conditions 20-1
20.3 Allowable Output Current Values 20-2
20.4 Internal Flash ROM Programming Conditions 20-2
20.5 DC Characteristics 20-3
20.5.1 DC Characteristics (V_{DD} = 4.5 to 5.5 V) 20-3
20.5.2 DC Characteristics (V_{DD} = 2.4 to 3.6 V) 20-5
20.6 AC Characteristics 20-7
20.6.1 AC Characteristics (V_{DD} = 4.5 to 5.5 V) 20-7
20.6.2 AC Characteristics (V_{DD} = 2.4 to 3.6 V) 20-15
20.7 A/D Converter Characteristics 20-23
20.7.1 A/D Converter Characteristics (V_{DD} = 4.5 to 5.5 V) 20-23
20.7.2 A/D Converter Characteristics (V_{DD} = 2.4 to 3.6 V) 20-23
20.8 D/A Converter Characteristics 20-25

Chapter 21 Special Function Registers (SFRs)

21.1 Overview 21-1
21.2 List of SFRs 21-1

Chapter 22 Package Dimensions 22-1

Chapter 1	Overview	1
Chapter 2	CPU Architecture	2
Chapter 3	CPU Control Functions	3
Chapter 4	Memory Control Functions	4
Chapter 5	Port Functions	5
Chapter 6	Clock Oscillation Circuit	6
Chapter 7	Time Base Counter (TBC)	7
Chapter 8	General-Purpose 8/16 Bit Timers	8
Chapter 9	Capture/Compare Timer	9
Chapter 10	Real-Time Counter (RTC)	10
Chapter 11	PWM Function	11
Chapter 12	Serial Port Functions	12
Chapter 13	A/D Converter Functions	13
Chapter 14	D/A Converter Functions	14
Chapter 15	Peripheral Functions	15
Chapter 16	External Interrupt Functions	16
Chapter 17	Interrupt Processing Functions	17
Chapter 18	Bus Port Functions	18
Chapter 19	Flash Memory	19
Chapter 20	Electrical Characteristics	20
Chapter 21	Special Function Registers (SFRs)	21
Chapter 22	Package Dimensions	22

Chapter 1

Overview

1. Overview

1.1 Overview

The MSM66577 family of highly functional CMOS 16-bit single chip microcontrollers utilize the nX-8/500S, Oki's proprietary CPU core.

Four channels of serial ports, consisting of two channels of synchronous serial ports with 32-byte FIFO registers and two channels of UART/synchronous serial ports, enable easy interfacing with external peripheral LSI devices such as an encoder/decoder or servocontroller.

A switching function permits selection of separate address and data lines or multiplexed lines for the bus interface to correspond to various peripheral LSI devices.

With features such as a dual clock function, programmable pull-up ports in which individual bits can be programmed, and a small, thin package, the MSM66577 family of microcontrollers is optimally suited for the system control of small-sized devices.

Reprogrammable Flash ROM versions (MSM66Q577LY/MSM66Q577) that operate on a single power supply (V_{DD} = 3.0 to 3.6 V/4.5 to 5.5 V) are available for sudden specification changes or version upgrades.

1.2 Features

The MSM66577 family has the following features.

- Instruction set with a wide variety of instructions
  - Super scalar instruction set
  - 8- and 16-bit arithmetic instructions
  - Multiply and divide instructions
    (High speed multiplier is not provided)
  - Bit manipulate instructions
  - Bit logical instructions
  - ROM table reference instructions

- Variety of addressing modes
  - Register addressing
  - Page addressing
  - Pointing register indirect addressing
  - Stack addressing
  - Immediate addressing

- Minimum instruction cycles
  - 67 ns at 30 MHz (4.5 to 5.5 V)
  - 143 ns at 14 MHz (2.4 to 3.6 V)
  - 61 µs at 32.768 kHz (2.4 to 3.6 V/4.5 to 5.5 V)

- Clock oscillation circuits
  - Main clock : 30 MHz (max.) crystal oscillator or ceramic resonator oscillator circuit
  - Subclock : 32.768 kHz crystal oscillator circuit

- Program memory (ROM)
  - Internal 128KB
  - External 1MB (in the case of $\overline{EA}$ pin activated)

- Data memory (RAM)
  - Internal 4KB
  - External 1020KB

- I/O ports
  - Input ports 8 ports (secondary function is an analog input port)
  - I/O ports 74 ports max. (with programmable pull-up resistors)

- Timers
  - Free-running counter 16 bits × 1
  - General-purpose auto reload timer 16 bits × 1, 8 bits × 1
  - 8-bit auto reload timer × 2
    Also functions as a 16-bit auto reload timer × 1
  - Baud rate generator and 8-bit auto reload timer × 3
  - Watchdog timer × 1
    Also functions as an 8-bit auto reload timer

- 8-bit PWM × 4

  (can also be used as two 16-bit PWMs)

- 8-bit serial ports
  - UART/Synchronous × 2
  - Synchronous, with 32-byte FIFO × 2

- A/D converter
  - 10-bit resolution, 8 channels

- D/A converter
  - 8-bit resolution, 2 channels

- Interrupts
  - Non-maskable: 1
  - Maskable: 8 external, 30 internal (29 vectors)
  - Three levels of priority

- ROM window function

- Standby modes
  - HALT mode
  - HOLD mode
  - STOP mode

- Package
  - 100-pin plastic TQFP (TQFP100-P-1414-0.50-K)
    (For external dimensions, refer to Chapter 22)

Table 1-1 MSM66577 Family of Products

Name	MSM66577L	MSM66577
Operating temperature range	–30°C to +70°C	
Power supply voltage/ maximum frequency	V_{DD} = 2.4 to 3.6 V/ f = 14 MHz	V_{DD} = 4.5 to 5.5 V/ f = 30 MHz
Minimum instruction execution time	143 ns at 14 MHz (2.4 to 3.6 V)	67 ns at 30 MHz (4.5 to 5.5 V)
	61 μs at 32.768 kHz (2.4 to 3.6/4.5 to 5.5 V)	
Internal ROM size (max. external)	128KB (1MB)	
Internal RAM size (max. external)	4KB (1MB)	
I/O ports	74 I/O pins (with programmable pull-up resistors) 8 input-only pins	
Timers	16-bit free running timer × 1 ch Compare out/capture input × 2 ch	
	16-bit timer (auto reload/timer out) × 1 ch 8-bit auto reload timer × 2 ch (can also be used as 16-bit timer × 1 ch) 8-bit auto reload timer × 1 ch	
	8-bit auto reload timer × 3 ch (also functions as serial communication baud rate generator)	
	Watchdog timer (also functions as 8-bit auto reload timer)	
	Watch timer (Real-time counter) × 1 ch	
	8-bit PWM × 4 ch (can also be used as 16-bit PWM × 2 ch)	
Serial port	UART/Synchronous × 2 ch Synchronous, with 32-byte FIFO × 2 ch	
A/D converter	10-bit A/D converter × 8	
D/A converter	8-bit D/A converter × 2	
External interrupt	Non-maskable × 1 ch Maskable × 8 ch	
Interrupt priority	3 levels	
Others	Separate address/data bus type/multiplexed bus type can be selected	
	Bus release function	
	Dual clocks	
FLASH ROM version	MSM66Q577LY (3.0 to 3.6 V)	MSM66Q577

1.3 Block Diagram

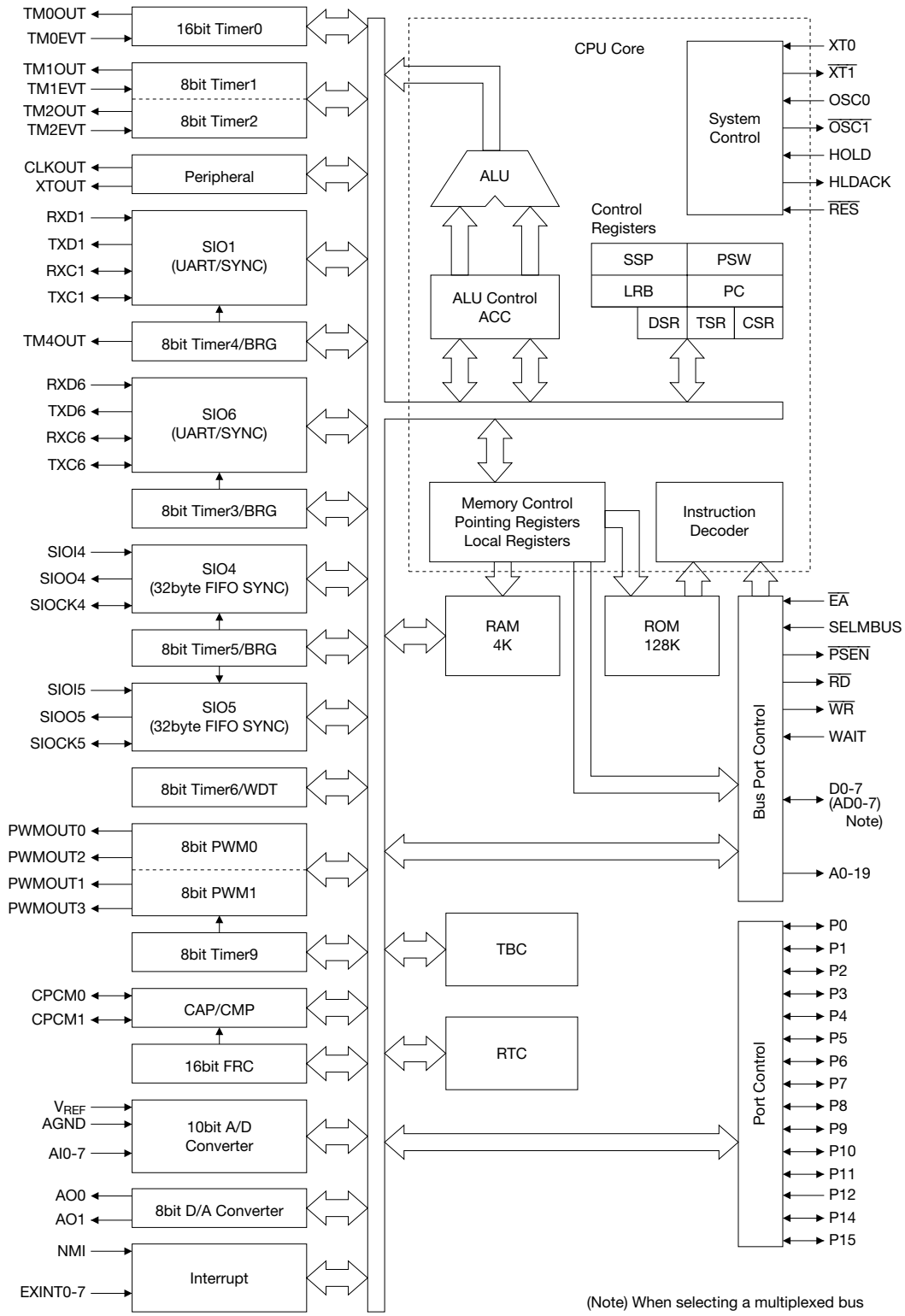
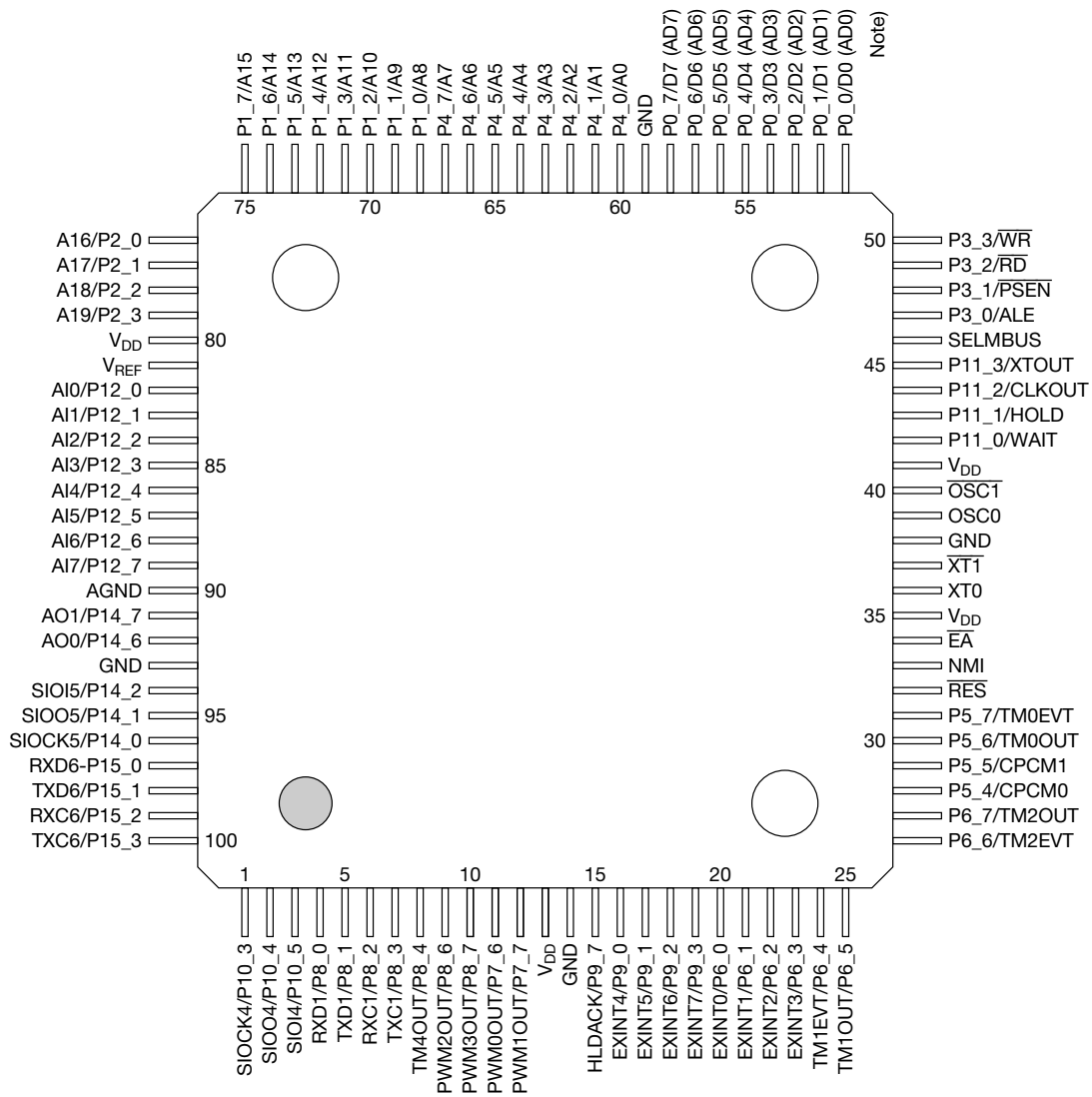


Figure 1-1 MSM66577 Family Block Diagram

1.4 Pin Configuration (Top View)



[Note] When selecting a multiplexed bus

Figure 1-2 MSM66577 Family Pin Configuration (100-pin TQFP package)

* For the external dimensions of the package, refer to Chapter 22, "Package Dimensions". For the connections of unused pins, refer to Section 1.5.3.

1.5 Pin Descriptions

1.5.1 Description of Each Pin

Table 1-2 lists the function of each pin in the MSM66577 family.
In the I/O column, "I" indicates an input pin, "O" indicates an output pin, and "I/O" indicates an I/O pin.

Table 1-2 Pin Descriptions (1/3)

Classification	Pin name	Function			
		I/O	Primary function	I/O	Secondary function
Port	P0_0/D0 (AD0) to P0_7/D7 (AD7)	I/O	8-bit I/O port 10 mA sink capability Pull-up resistors can be specified for each individual bit	I/O	External memory access Data I/O port (Address output/data I/O port when a multiplexed bus is selected)
	P1_0/A8 to P1_7/A15	I/O	8-bit I/O port Pull-up resistors can be specified for each individual bit	O	External memory access Address output port
	P2_0/A16 to P2_3/A19	I/O	4-bit I/O port Pull-up resistors can be specified for each individual bit	O	External memory access Address output port
	P3_0/ALE	I/O	4-bit I/O port 10 mA sink capability	O	External memory access Address latch enable signal output pin
	P3_1/$\overline{\text{PSEN}}$		Pull-up resistors can be specified for each individual bit	O	External program memory access Read strobe output pin
	P3_2/$\overline{\text{RD}}$			O	External memory access Read strobe output pin
	P3_3/$\overline{\text{WR}}$			O	External memory access Write strobe output pin
	P4_0/A0 to P4_7/A7	I/O	8-bit I/O port Pull-up resistors can be specified for each individual bit	O	External memory access Address output port (when a separate bus is selected)
	P5_4/CPCM0	I/O	4-bit I/O port Pull-up resistors can be specified for each individual bit	I/O	Capture 0 input/ Compare 0 output pin
	P5_5/CPCM1			I/O	Capture 1 input/ Compare 1 output pin
	P5_6/TM0OUT			O	Timer 0 timer output pin
	P5_7/TM0EVT			I	Timer 0 external event input pin
	P6_0/EXINT0	I/O	8-bit I/O port Pull-up resistors can be specified for each individual bit	I	External interrupt 0 input pin
	P6_1/EXINT1			I	External interrupt 1 input pin
	P6_2/EXINT2			I	External interrupt 2 input pin
	P6_3/EXINT3			I	External interrupt 3 input pin
	P6_4/TM1EVT			I	Timer 1 external event input pin
	P6_5/TM1OUT			O	Timer 1 timer output pin
	P6_6/TM2EVT			I	Timer 2 external event input pin
	P6_7/TM2OUT			O	Timer 2 timer output pin

Table 1-2 Pin Descriptions (2/3)

Classification	Pin name	Function			
		I/O	Primary function	I/O	Secondary function
Port	P7_6/PWM0OUT	I/O	2-bit I/O port	O	PWM0 output pin
	P7_7/PWM1OUT		Pull-up resistors can be	O	PWM1 output pin
			specified for each individual bit		
	P8_0/RXD1	I/O	7-bit I/O port	I	SIO1 receive data input pin
	P8_1/TXD1		Pull-up resistors can be	O	SIO1 transmit data output pin
	P8_2/RXC1		specified for each individual bit	I/O	SIO1 receive clock I/O pin
	P8_3/TXC1			I/O	SIO1 transmit clock I/O pin
	P8_4/TM4OUT			O	Timer 4 timer output pin
	P8_6/PWM2OUT			O	PWM2 output pin
	P8_7/PWM3OUT			O	PWM3 output pin
	P9_0/EXINT4	I/O	5-bit I/O port	I	External interrupt 4 input pin
	P9_1/EXINT5		Pull-up resistors can be	I	External interrupt 5 input pin
	P9_2/EXINT6		specified for each individual bit	I	External interrupt 6 input pin
	P9_3/EXINT7			I	External interrupt 7 input pin
	P9_7/HLDACK			O	HOLD mode output pin
	P10_3/SIOCK4	I/O	3-bit I/O port	I/O	SIO4 transmit-receive clock I/O pin
	P10_4/SIOO4		Pull-up resistors can be	I	SIO4 receive data input pin
	P10_5/SIOI4		specified for each individual bit	O	SIO4 transmit data output pin
	P11_0/WAIT	I/O	4-bit I/O port	I	External data memory access
			10 mA sink capability		wait input pin
	P11_1/HOLD		Pull-up resistors can be	I	HOLD mode request input pin
	P11_2/CLKOUT		specified for each individual bit	O	Main clock pulse output pin
	P11_3/XTOUT			O	Subclock pulse output pin
	P12_0/AI0	I	8-bit input port	I	A/D converter analog input port
	to				
	P12_7/AI7				
	P14_0/SIOCK5	I/O	5-bit I/O port	I/O	SIO5 transmit-receive clock I/O pin
	P14_1/SIOO5		Pull-up resistors can be	O	SIO5 transmit data output pin
	P14_2/SIOI5		specified for each individual bit	I	SIO5 receive data input pin
	P14_6/AO0			O	D/A converter analog output port
	P14_7/AO1			O	D/A converter analog output port
	P15_0/RXD6	I/O	4-bit I/O port	I	SIO6 receive data input pin
	P15_1/TXD6		Pull-up resistors can be	O	SIO6 transmit data output pin
	P15_2/RXC6		specified for each individual bit	I/O	SIO6 receive clock I/O pin
	P15_3/TXC6			I/O	SIO6 transmit clock I/O pin

Table 1-2 Pin Descriptions (3/3)

Classification	Pin name	I/O	Function
Power supply	V_{DD}	I	Power supply pin Connect all V_{DD} pins to the power supply. *
	GND	I	GND pin Connect all GND pins to GND. *
	V_{REF}	I	Analog reference voltage pin
	AGND	I	Analog GND pin
Oscillation	XT0	I	Subclock oscillation input pin Connect to a crystal oscillator of f = 32.768 kHz.
	$\overline{XT1}$	O	Subclock oscillation output pin Connect to a crystal oscillator of f = 32.768 kHz. The clock output is opposite in phase to XT0.
	OSC0	I	Main clock oscillation input pin Connect to a crystal or ceramic oscillator. Or, input an external clock.
	$\overline{OSC1}$	O	Main clock oscillation output pin Connect to a crystal or ceramic oscillator. The clock output is opposite in phase to OSC0. Leave this pin unconnected when an external clock is used.
Reset	$\overline{RES}$	I	Reset input pin
Others	NMI	I	Non-maskable interrupt input pin
	$\overline{EA}$	I	External program memory access input pin If the $\overline{EA}$ pin is enabled (low level), the internal program memory is masked and the CPU executes the program code in external program memory through all address space.
	SELMBUS	I	External bus interface setting input pin SELMBUS = H: Address/data separate bus type SELMBUS = L: Multiplexed bus type

* Each of the family devices has unique pattern routes for the internal power and ground. Connect the power supply voltage to all V_{DD} pins and the ground potential to all GND pins. If a device may have one or more V_{DD} or GND pins to which the power supply voltage or the ground potential is not connected, it cannot be guaranteed for normal operation.

1.5.2 Pin Configuration

A simplified pin configuration for each pin of the MSM66577 family is shown in Table 1-3 and Figure 1-3.

Table 1-3 Configuration of Each Pin

Pin name	Type	Pin name	Type
P0_0 to P0_7	6	P9_0 to P9_3, P9_7	5
P1_0 to P1_7	5	P10_3 to P10_5	5
P2_0 to P2_3	5	P11_0 to P11_3	5
P3_0, P3_1	4	P12_0 to P12_7	3
P3_2, P3_3	5	P14_0 to P14_2	5
P4_0 to P4_7	5	P14_6, P14_7	7
P5_4 to P5_7	5	P15_0 to P15_3	5
P6_0 to P6_7	5	$\overline{\text{RES}}$	2
P7_6, P7_7	5	SELMBUS, NMI, $\overline{\text{EA}}$	1
P8_0 to P8_4, P8_6, P8_7	5		

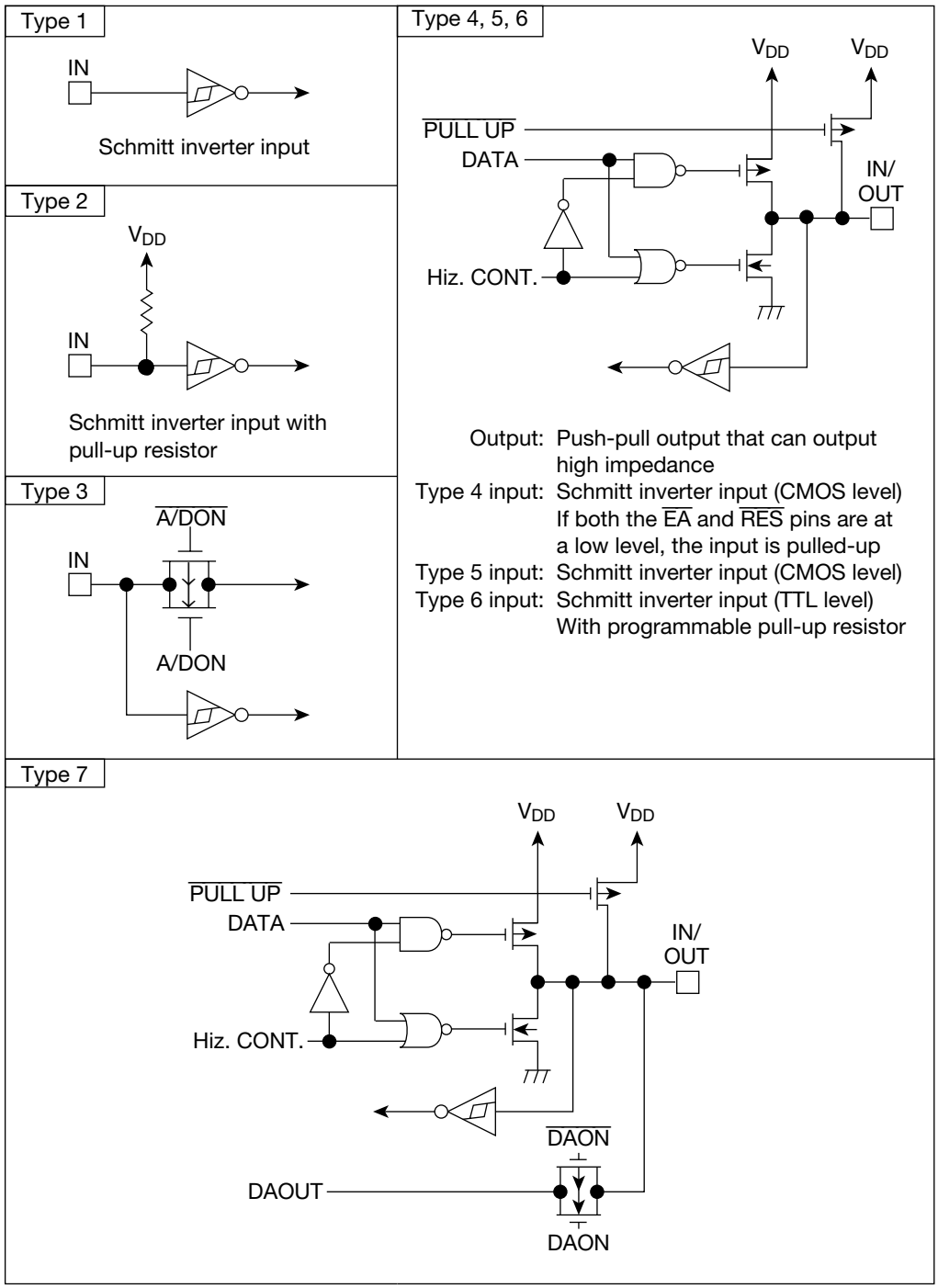


Figure 1-3 Types of Pin Configurations

1.5.3 Connections for Unused Pins

Table 1-4 lists the pin connections for unused pins.

Table 1-4 Connections for Unused Pins

Pin	Pin connection
P0_0 to 0_7	When a programmable pull-up resistor is set: Open When input is set: High or Low level When output is set: Open
P1_0 to 1_7	
P2_0 to 2_3	
P3_0 to 3_3	
P4_0 to 4_7	
P5_4 to 5_7	
P6_0 to 6_7	
P7_6, P7_7	
P8_0 to P8_4, P8_6, P8_7	
P9_0 to 9_3, P9_7	
P10_3 to 10_5	
P11_0 to 11_3	
P14_0 to 14_2, P14_6, P14_7	
P15_0 to 15_3	
P12_0 to 12_7	VREF or AGND
VREF	V_{DD}
AGND	GND
NMI	High or Low level
$\overline{EA}$	High level
XT0	GND*
$\overline{OSC1}$, $\overline{XT1}$	Open
SELMBUS	GND or V_{DD} (Note)

* If the subclock (XT0, $\overline{XT1}$) is not used, in addition to connecting the XT0 pin to GND and leaving XT1 unconnected, the peripheral control register (PRPHCON) must be set. For details refer to Chapter 6, "Clock Oscillation Circuit."

[Note] For SELMBUS,
0: Multiplexed bus
1: Separate bus

1.6 Basic Operational Timing

The MSM66577 family is configured such that one pulse of the main clock (CLK) is one state. In other words, one state is 33.3 ns (at 30 MHz). One instruction cycle consists of more than one state (S1, S2, ..., Sn).

The number of states required for program execution differs depending upon the instruction. The minimum is 2 states and the maximum is 48 states. (For details, refer to the nX-8/500S Core Instruction Manual.)
To achieve high-speed execution of instructions, one byte of the instruction is pre-fetched. While one instruction is being executed, the next instruction will be fetched.

Figure 1-4 through Figure 1-7 show basic timing examples.

If program memory is accessed externally, a number of wait cycles (0 to 3 cycles) specified by the ROM ready control register (ROMRDY) are inserted. If data memory is accessed externally, 2 or 3 cycles (1 cycle = 1 state) are automatically inserted for a 1 byte read or write. In addition, the number of wait cycles (0 to 7 cycles) specified by the RAM ready control register (RAMRDY) will also be inserted.

For external memory access timings, refer to Chapter 18, "Bus Port Functions."

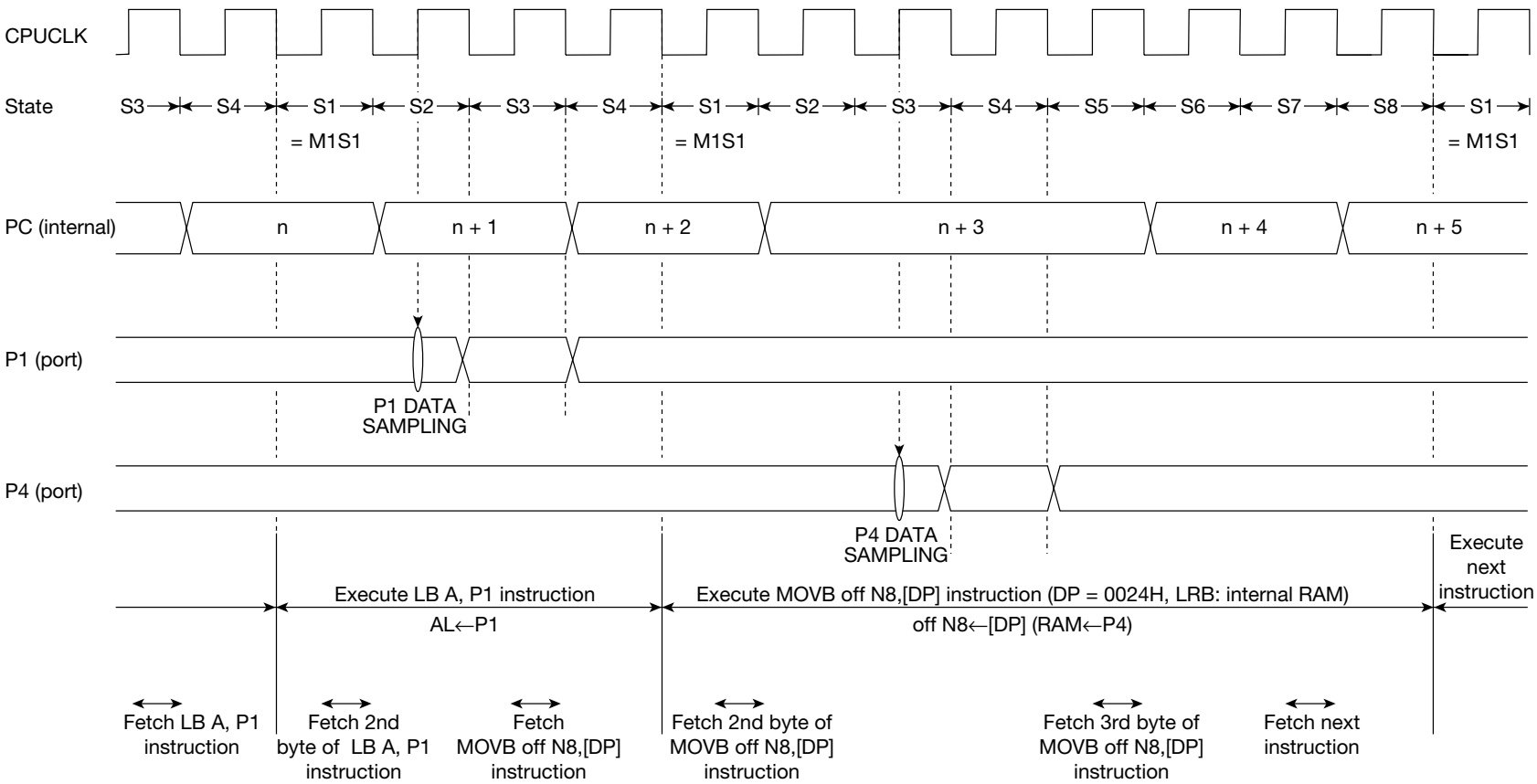


Figure 1-4 Basic Operation Timing Example (Reading Port Data)

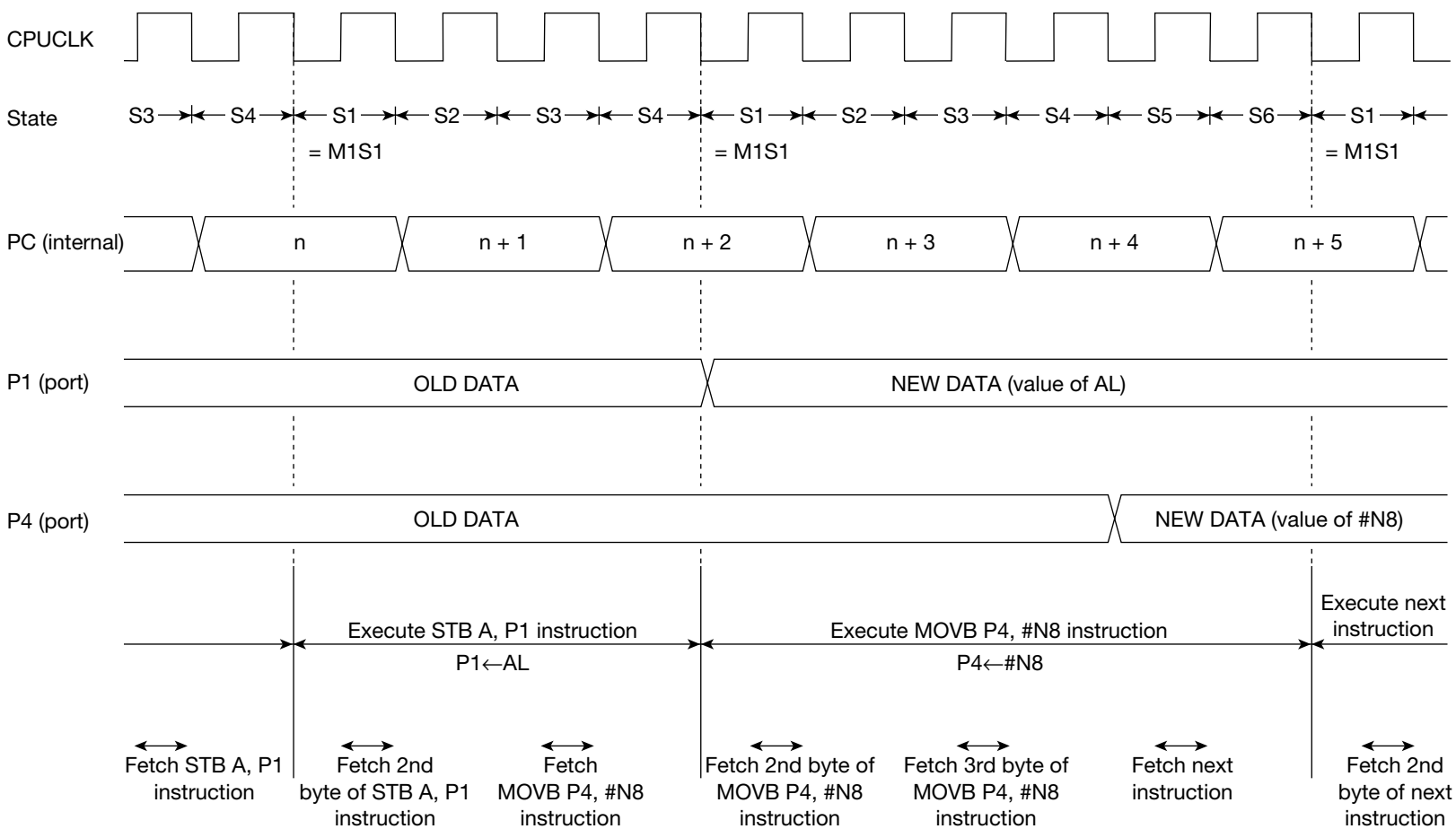
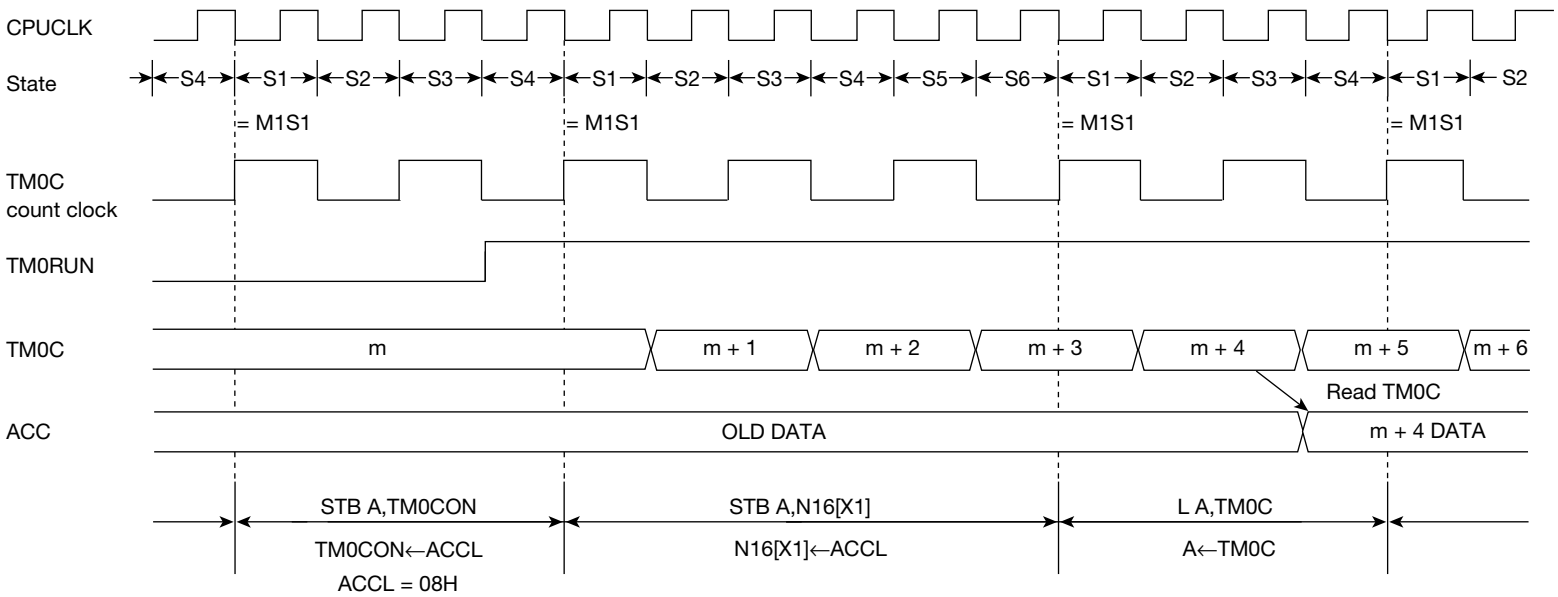


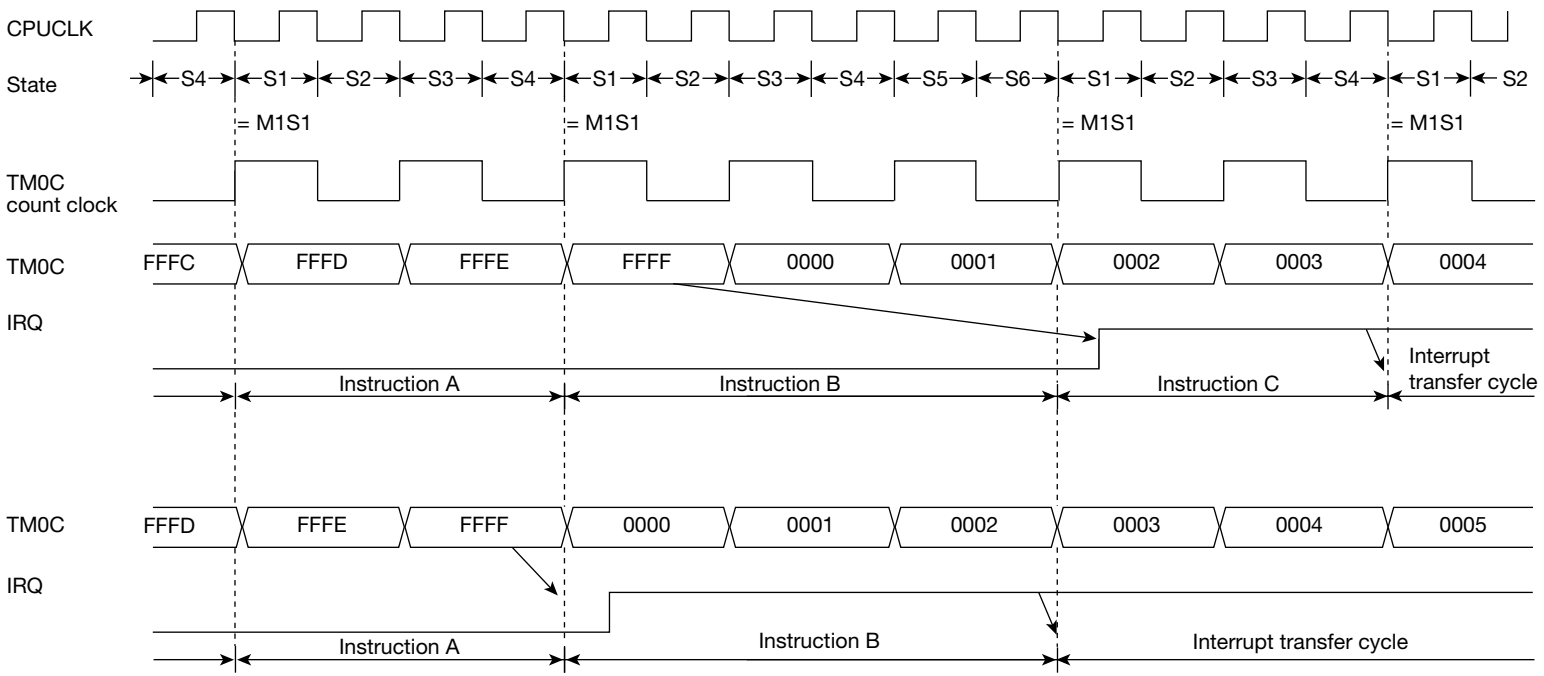
Figure 1-5 Basic Operation Timing Example (Writing Port Data)



[Note]

- The timing when the TM0RUN bit becomes "1" differs depending upon the instruction used.
- The timing for reading TM0C differs depending upon the instruction used.
- The TM0C count timing differs depending upon the selected TM0C clock.

Figure 1-6 Timer 0 Operation Timing Example



[Note]

- There are 14 interrupt transfer cycles. However, if the program memory space has been extended 1MB, then there will be 17 cycles.
- IRQ is reset to "0" at the 3rd interrupt transfer cycle.

Figure 1-7 Interrupt Transfer Timing Example

Chapter 2

CPU Architecture

2. CPU Architecture

2.1 Overview

The MSM66577 microcontroller family utilize the nX-8/500S, Oki's proprietary 16-bit CPU core.
The nX-8/500S performs various operations mainly by using an accumulator and register set. Almost all instructions and addressing modes are applicable both to byte-format and word-format data. And it also has bit processing functions.

Program memory space and data memory space are separated and provided respectively. Each can be expanded up to 1MB. In addition, special dedicated addressing modes are provided for some specific portion of data space such as Special Function Registers area, fixed page area, and current page area and so on, for the purpose of efficient programming.

For further details, refer to the "nX-8/500S CPU Core Instruction Manual".

2.2 Memory Space

Program memory space and data memory space are set independently. At reset, up to 64 KB (max.) can be accessed for each. By changing settings of the memory size control register (MEMSCON), located in the SFR area, the program memory space and the data memory space can each be expanded up to 1MB.

2.2.1Memory Space Expansion

The memory size control register (MEMSCON) is located in the SFR register and specifies the size of the memory space. The program memory space can be expanded to 1MB by setting the LROM bit (bit 1) to "1". The data memory space can be expanded to 1MB by setting the LRAM bit (bit 0) to "1".

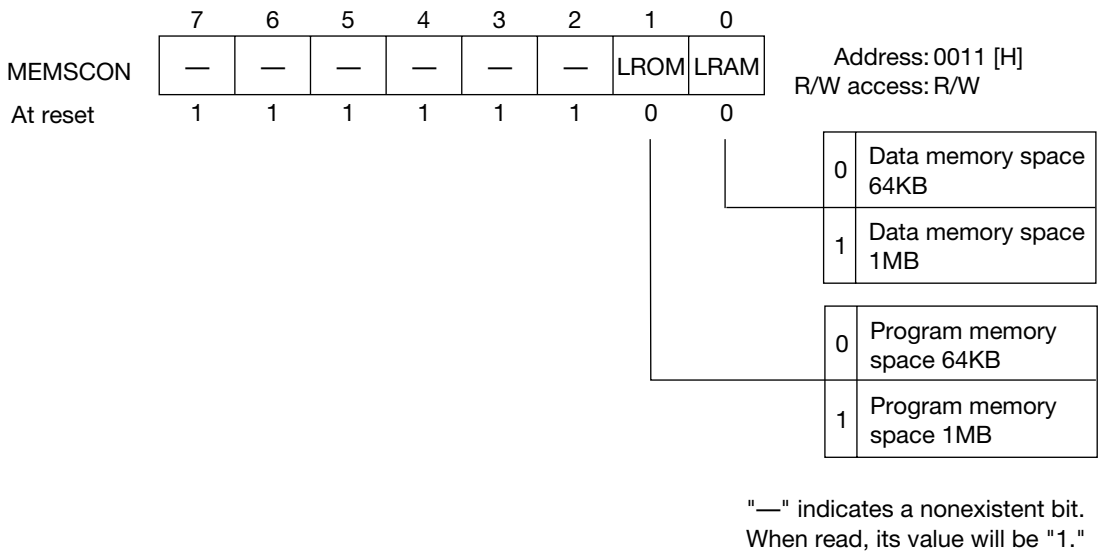


Figure 2-1 MEMSCON Configuration

To write to the LROM bit of MEMSCON, write "5H" and then "0AH" to the upper 4 bits of the memory size acceptor (MEMSACP) register located in the SFR area. Likewise, to write to the LRAM bit of MEMSCON, first write "5H" and then "0AH" to the lower 4 bits of the memory size acceptor (MEMSACP).

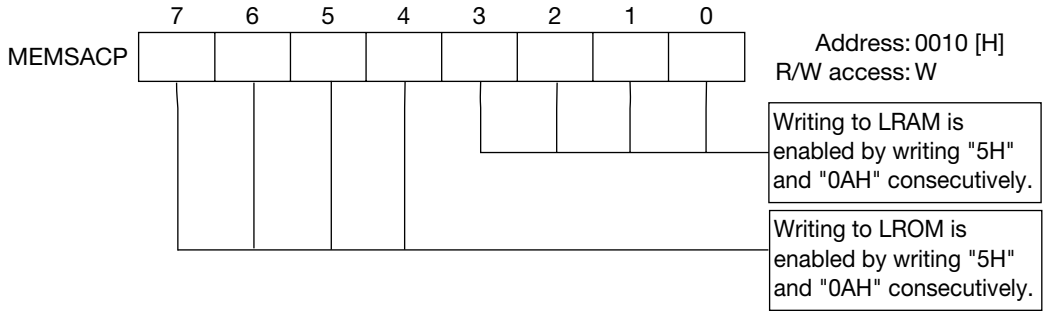


Figure 2-2 MEMSACP Configuration

Note : If the FJ, FCAL, or FRT instruction is executed while the LROM bit is being reset to "0", the op code trap is generated and system reset will be executed.

If the LROM or LRAM bits are set to "1", the memory space expansion is actually enabled after execution of the instruction that follows the LROM or LRAM bit write instruction. Programming examples to expand the program memory space are listed below.

- SMALL memory space (64KB program memory space, 64KB data memory space)
  MOVB MEMSACP, #05H
  MOVB MEMSACP, #0AH
  MOVB MEMSCON, #00H (initial value)

- COMPACT memory space (64KB program memory space, 1MB data memory space)
  MOVB MEMSACP, #05H
  MOVB MEMSACP, #0AH
  MOVB MEMSCON, #01H

- MEDIUM memory space (1MB program memory space, 64KB data memory space)
  MOVB MEMSACP, #50H
  MOVB MEMSACP, #0A0H
  MOVB MEMSCON, #02H

- LARGE memory space (1MB program memory space, 1MB data memory space)
  MOVB MEMSACP, #55H
  MOVB MEMSACP, #0AAH
  MOVB MEMSCON, #03H

MEMSCON can be written only once after reset (due to a $\overline{\text{RES}}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap). Therefore, to change the memory space model once set to the other, reset and write again to the MEMSCON.

2.2.2 Program Memory Space

The program memory space is also called "ROM space".

The MSM66577 family can access a maximum of 1MB (1,048,576 bytes) of program memory in 64KB (65,536 bytes) unit segments from segment 0 to 15. However, if more than 64KB (segments 1 to 15) are to be accessed, the LROM bit of the MEMSCON (memory size control register) SFR must be set to "1".

The code segment register (CSR) specifies the segment to be used, and the program counter (PC) specifies the address in the segment. However, the segment to be used in the execution of ROM table reference instructions (such as LC A, obj) and the ROM window function is specified by the table segment register (TSR).

The 128KB (131,072 bytes) area in segments 0 and 1constitutes the internal ROM area and the 64KB areas in segments 2 to 15 form the external ROM area.

The following areas are assigned to segment 0:

- Vector table area (84 bytes)
- VCAL table area (32 bytes)

In addition, the following area is assigned to each segment.

- ACAL area (2,048 bytes)

Figure 2-3 shows a memory map of the program memory space.

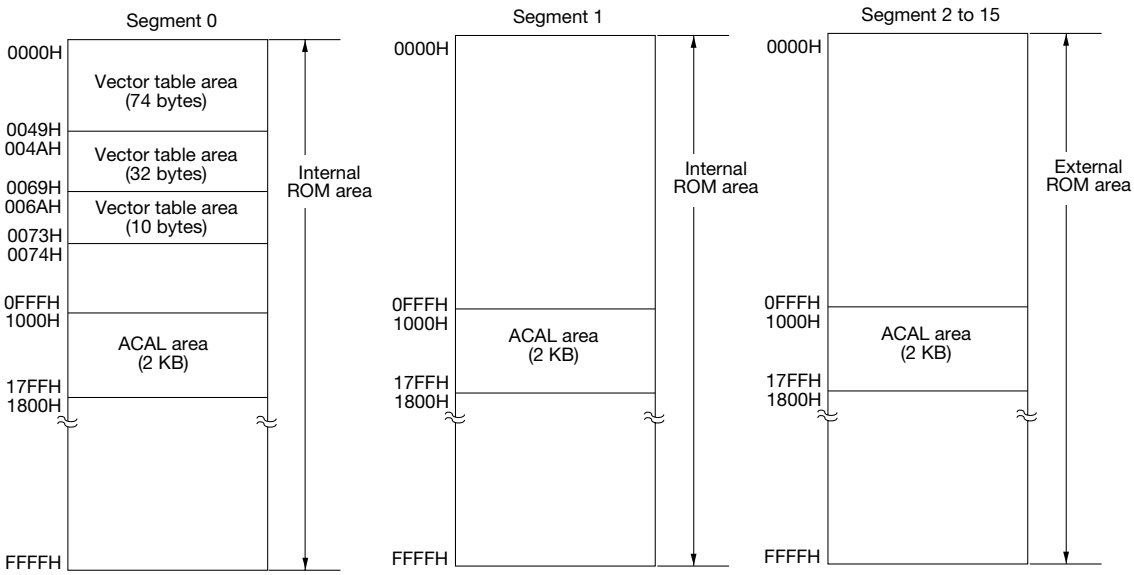


Figure 2-3 Memory Map of Program Memory Space

(1) Accessing program memory space

Program memory space is accessed by the program counter (PC) and the code segment register (CSR). However, when a ROM table reference instruction (such as LC A, obj) or a ROM window function (refer to Section 4.3) is executed, program memory space is accessed according to the contents of the table segment register (TSR) and the register specified by the instruction.

Access of the internal ROM area and the external memory area of the program memory space is automatically switched by internal device operation depending on the status of the $\overline{EA}$ pin and program addresses.

When a high level is input to the $\overline{EA}$ pin, the internal ROM area is accessed if the program address is between 0000H and 1FFFFH, and the external ROM area is accessed if the address is between 20000H to FFFFFH. When the external ROM area is to be accessed, the secondary functions of the external memory control pins (port 0, 1, 2, 3 and 4) must be set.

The area from 0000H to 1FFFDH can be fetched by the internal program. Be careful that the final address of instruction code does not exceed 1FFFDH. The final address of the table data is 1FFFFH.

When a low level is input to the $\overline{EA}$ pin, the external program memory area is accessed for all program addresses.

If the external memory area of the program memory space is accessed in a separate bus type, Port 0 (data input), Port 4 (addresses A0 to A7 outputs), Port 1 (addresses A8 to A15 outputs) and Port 2 (addresses A16 to A19 outputs) operate as bus ports, and the P3_1/$\overline{PSEN}$ pin becomes active.

If the external memory area of the program memory space is accessed in a multiplexed bus type, Port 0 (address output and data input),Port 1 (addresses A8 to A15 outputs) and Port 2 (addresses A16 to A19 outputs) operate as bus ports, and P3_0/ALE pin and P3_1/$\overline{PSEN}$ pin become active.

(2) Vector table area

The 74-byte area of addresses from 0000H to 0049H and the 10-byte area of addresses from 006AH to 0073H in segment 0 of program memory space are used as the vector table area that stores branch addresses for all types of resets and interrupts (29 types) as shown in Table 2-1.

If a reset or interrupt occurs, the corresponding 2-byte branch address, stored in the vector table, is loaded into the PC. (The even address contains the lower order data and the odd address contains the upper order data.) At the same time, "0" is loaded into the Code Segment Register (CSR) and program execution starts from the loaded segment 0 address. Therefore, if a reset or interrupt occurs during execution of an instruction in segment 1 (or segment other than 0), program control will branch to an address in segment 0.

With reasons described above, reset routine and interrupt routines must be located in segment 0. This fact is important for medium and large memory model programming. Proper alignment attribute must be applied to your relocatable interrupt routines.

In medium and large memory model you specified by MEMSCON setting, CPU automatically provides extra stack area for the CSR contents. When RTI instruction is executed, CSR contents in stack are re-stored into CSR and program execution is continued in the same program segment.

If this area is not used as a vector table area, it can be used as a normal program area.

Table 2-1 lists the vector table addresses for each type of reset or interrupt.

[Example] Program starting address of 0200H due to $\overline{\text{RES}}$ pin input

Program address	Data code	
0000H	00H	(lower order data for program start address)
0001H	02H	(upper order data for program start address)

Table 2-1 Vector Table List

Vector table starting address [H]	Interrupt or reset factor
0000	Reset by $\overline{\text{RES}}$ pin input
0002	Reset by execution of BRK instruction
0004	Reset by overflow of watchdog timer
0006	Reset by opcode trap
0008	Interrupt by NMI pin input (non-maskable interrupt)
000A	Interrupt by EXINT0 pin input (external interrupt 0)
000C	Interrupt by overflow of free running counter
0016	Interrupt by CPCM0 event input, compare match
0018	Interrupt by CPCM1 event input, compare match
001A	Interrupt by overflow of timer 0
001C	Interrupt by EXINT1 pin input (external interrupt 1)
001E	Interrupt by EXINT2 pin input (external interrupt 2)
0020	Interrupt by EXINT3 pin input (external interrupt 3)
0022	Interrupt by overflow of timer 1
0024	Interrupt by overflow of timer 2
0026	Interrupt by overflow of timer 3
0028	Interrupt by SIO0 transmit buffer empty, transmit completion, receive completion
002A	Interrupt by EXINT4 pin input (external interrupt 4)
002C	Interrupt by EXINT5 pin input (external interrupt 5)
002E	Interrupt by EXINT6 pin input (external interrupt 6)
0030	Interrupt by EXINT7 pin input (external interrupt 7)
0036	Interrupt by overflow of timer 4
0038	Interrupt by SIO1 transmit buffer empty, transmit completion, receive completion
003A	Interrupt by overflow of timer 5
003C	Interrupt by SIO5 transfer completion
003E	Interrupt by SIO6 transmit buffer empty, transmit completion, receive completion
0040	Interrupt by SIO4 transfer completion
0042	Interrupt by overflow of timer 6
0044	Interrupt by A/D conversion scan channel cycle completion/select mode completion
0048	Interrupt by real-time counter output (interval: 0.125 to 1 s)
006A	Interrupt by PWC0 overflow, PWC0 and PWR0 match
006C	Interrupt by PWC1 overflow, PWC1 and PWR1 match
006E	Interrupt by PWC0 and PWR2 match
0070	Interrupt by PWC1 and PWR3 match
0072	Interrupt by overflow of timer 9

(3) VCAL table area

The VCAL table area is assigned to the 32-byte area of program memory space in segment 0 from address 004AH to 0069H and stores branch addresses for 1-byte call instructions (VCAL: 16 types).

If a VCAL instruction is executed, the next address after the VCAL instruction is saved onto the system stack, the system stack pointer (SSP) is decremented by 2, and the corresponding 2-byte address stored in the vector table is loaded into the PC. (The even address contains the lower data and the odd address contains the upper data). The program begins execution from the loaded address.

However, if the program memory space has been expanded to 1MB, the SSP is decremented by 4 because the CSR value is also saved at the same time that the PC is saved. Also, the CSR is loaded with "0" at the same time as the branch address is loaded into the PC. Therefore, if a VCAL instruction is executed in segment 1, program control will branch to a branch address in segment 0.

If the program memory space is up to 64KB (the LROM bit of MEMSCON is "0"), execution of a RT instruction will return program control from the subroutine branched to by the VCAL instruction. If the program memory space is 1MB (the LROM bit is "1"), execution of a FRT instruction returns program control from the subroutine branched to by the VCAL instruction.

If this area is not used as the VCAL table area, it can be used as a normal program area.

Table 2-2 lists the VCAL vector addresses.

[Example] Program starting address of 0400H due to VCAL 4AH instruction

Program address	Data code	
004AH	00H	(lower order data for subroutine start address)
004BH	04H	(upper order data for subroutine start address)

Table 2-2 VCAL Vector Address List

VCAL table starting address [H]	VCAL instruction
004A	VCAL 4AH
004C	VCAL 4CH
004E	VCAL 4EH
0050	VCAL 50H
0052	VCAL 52H
0054	VCAL 54H
0056	VCAL 56H
0058	VCAL 58H
005A	VCAL 5AH
005C	VCAL 5CH
005E	VCAL 5EH
0060	VCAL 60H
0062	VCAL 62H
0064	VCAL 64H
0066	VCAL 66H
0068	VCAL 68H

(4) ACAL area

The 2KB area from 1000H to 17FFH of each program segment is called ACAL area. The subroutines located in this area can be called by 2-byte call instruction (ACAL).
ACAL is an in-segment call instruction which does not rewrite the CSR contents.

If an ACAL instruction is executed, the address following the next address after the ACAL instruction is saved onto the system stack, the system stack pointer (SSP) is decremented by 2, and 11-bit data included in the ACAL instruction code is loaded into the PC. Program execution begins at the loaded address (1000H to 17FFH).

2.2.3 Data Memory Space

A maximum of 1MB (1,048,576 bytes) of data memory can be accessed.

The following areas are assigned to the data memory space: a special function register area (SFR: 256 bytes), a reserved area (256 bytes), a fixed page area (FIX: 256 bytes), an internal RAM area (4,096 bytes), a local register setting area (2,048 bytes) and an external memory area (1,043,968 bytes).

A pointing register area (PR: 64 bytes) and a special bit addressing area (sbafix: 64 bytes) are assigned to the fixed page area. The ROM window setting area (1200H to 0FFFFH of segment 0 and 1000H to 0FFFFH of segments 1 to 15) is assigned to the external data memory area.

So that data can be exchanged between two or more data segments without changing DSR, there is a common area that starts at data memory address 0H. The SFR area, reserved area and fixed page area always belong to the common area.

Figure 2-4 shows a memory map of the data memory space.

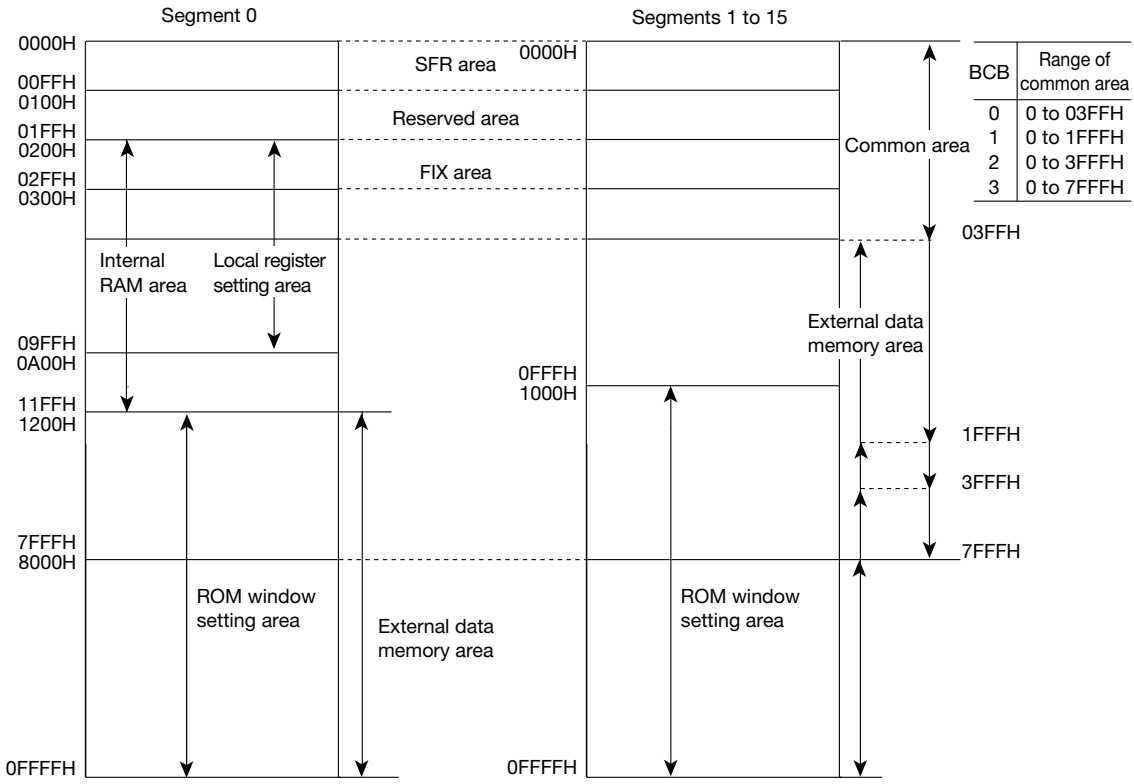


Figure 2-4 Memory Map of Data Memory Space

(1) Special function register (SFR) area

The group of registers with special functions such as mode registers for internal peripheral hardware, control registers and counters are assigned to the 256-byte area in data memory space from 0000H to 00FFH. Refer to Chapter 21, "Special Function Registers (SFRs)" for a more detailed description.

(2) Reserved area

The 256-byte data memory space from 0100H to 01FFH is reserved for future use as an expanded SFR area. This area cannot be used.

(3) Internal RAM area

Internal RAM is assigned to the 4,096-byte area in data memory space from 0200H to 11FFH.

(4) Fixed page (FIX) area

A pointing register (PR) area and a special bit addressing (sbafix) area are assigned to the 256-byte area in data memory from 0200H to 02FFH.

The pointing register area is assigned to addresses 0200H to 023FH and contains 8 sets of the following 4 registers.

- Index register (X1, X2)
- Data pointer (DP)
- User stack pointer (USP)

All of the above are 16-bit registers. Even addresses contain lower order data and odd addresses contain higher order data.

The special bit address area is assigned to addresses 02C0H to 02FFH. SB, RB, JBR and JBS instructions to this area can be implemented in a small number of bytes.

Figure 2-5 shows the map of the fixed page area.

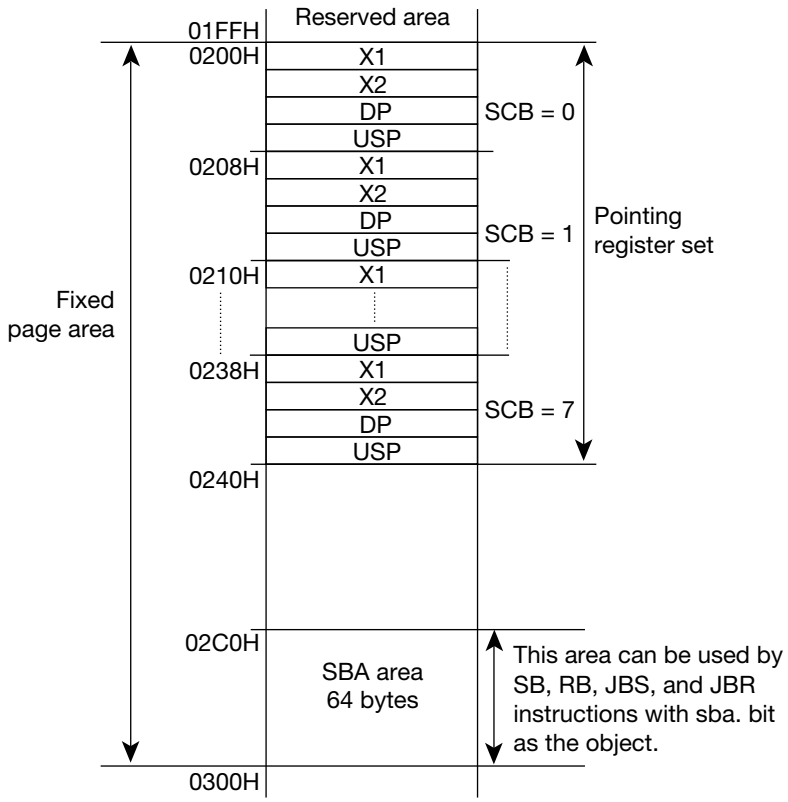


Figure 2-5 Map of Fixed Page Area

(5) Local register setting area

The local register setting area is the 2KB area of data memory from 0200H to 09FFH. Local registers are set in 8-byte units, as specified by the lower 8 bits of LRB (LRBL).

Figure 2-6 shows the map of the local register setting area.

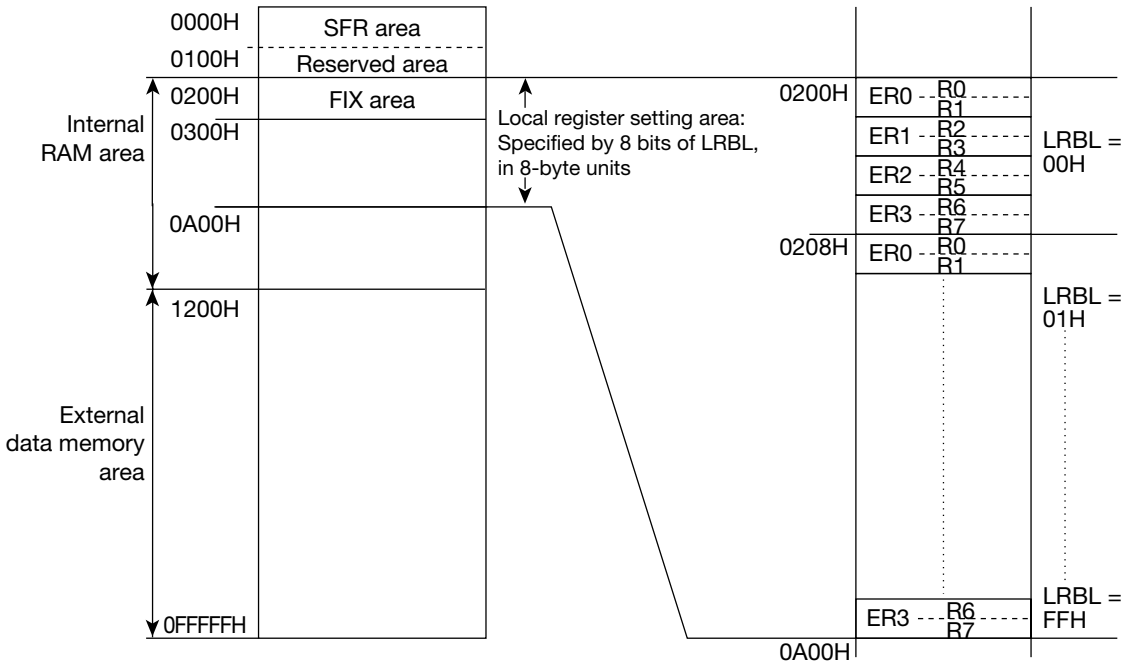


Figure 2-6 Map of Local Register Setting Area

(6) External data memory area

The external data memory area is the 1,043,968-byte area of data memory from 1200H to 0FFFFFH (Note 1). If this external data memory is to be accessed, the secondary functions of memory related pins (ports 0, 1, 2, 3 and 4) must be set. In a separate bus type, the external data memory is accessed by Port 0 (data I/O: D0 to D7), Port 4 (address output: A0 to A7), Port 1 (address output: A8 to A15), Port 2 (address output: A16 to A19), P3_3/$\overline{WR}$ (write strobe output function) and P3_2/$\overline{RD}$ (read strobe output function) signals.

In a multiplexed bus type, the external data memory is accessed by Port 0 (combined I/O of data input and address output: AD0 to AD7), Port 1 (address output: A8 to A15), Port 2 (address output: A16 to A19), P3_0/ALE (address latch enable output function), P3_2/$\overline{RD}$ (read strobe output function) and P3_3/$\overline{WR}$ (write strobe output function) signals.

The 1,043,968-byte area from 1200H to 0FFFFFH of data memory is the external data memory area. However, the ROM window function can be set by the ROM window setting register. If the ROM window function is used in the specified area (address 1200H and above), instead of accessing data in the data memory space, instructions (read operations) will access data in the program memory space at the same address.

The ROM window function is valid if the register (ROMWIN) that enables the ROM window function is set and the accessed (read) address is in external data memory.

(7) Common area

There is a common area in the data memory space to enable the exchange of data between two or more data segments. The common area is located at the bottom of data memory, beginning at offset address 0H in each segment. The range of the common area is determined by the value of BCB in the PSW.

B C B		BCB range of common area
1	0	
0	0	0H to 03FFH
0	1	0H to 1FFFH
1	0	0H to 3FFFH
1	1	0H to 7FFFH

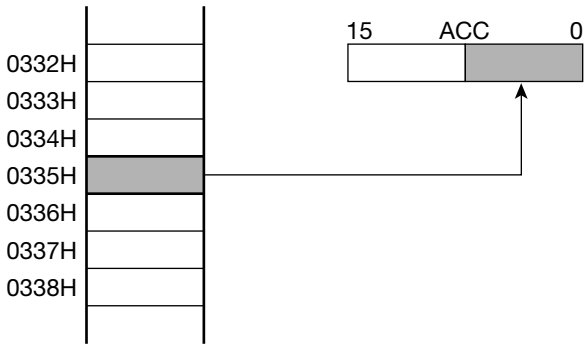
2.2.4 Data Memory Access

Examples of memory access are presented below for the cases when an instruction performs a byte operation and a word operation in the data memory space.

(1) Byte operations

In the case of a byte operation, the address obtained from the instruction points to the targeted 8-bit data.

[Example] LB A, [DP]: where the contents of DP are 0335H



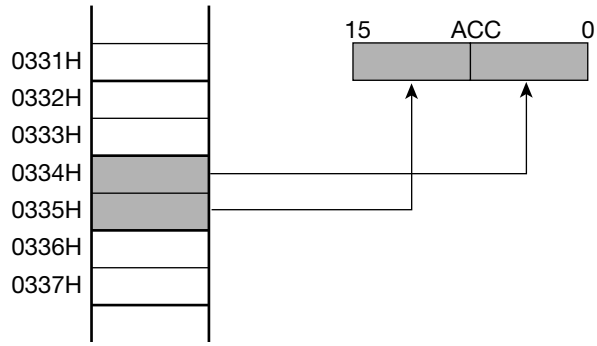
(2) Word operations

In the case of a word operation, corresponding to the address obtained from the instruction, the address with least significant bit (LSB) set to "0" (even address) points to the lower order 8-bit data and the address with LSB set to "1" (odd address) points to the upper order 8-bit data to form the targeted 16-bit data.

Therefore, a targeted 16-bit data formed with upper oder 8-bit data for the odd address and lower oder 8-bit for the even address can not be accessed. (The boundary exists between two bytes in word operation.)

Yet such a boundary limit does not exist for the program memory space.

[Example] L A, [DP]: where the contents of DP are 0334H (or 0335H)



2.3 Registers

Registers are classified by function as the arithmetic register, control registers, pointing registers, special function registers, local registers and segment registers.

Figure 2-7 shows the configuration of each register.

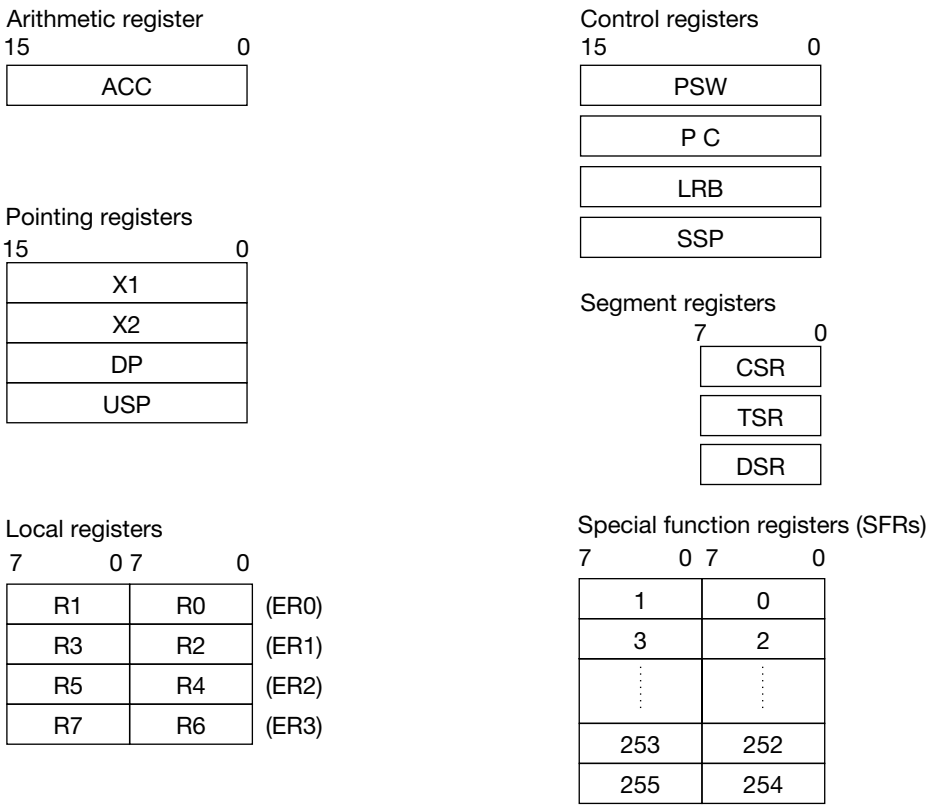


Figure 2-7 Register Configurations

2.3.1 Arithmetic Register (ACC)

The arithmetic register is a 16-bit accumulator (ACC), central to all type of arithmetic operations.

If a transfer or arithmetic operation is:

- a word operation, all 16 bits (bits 15 to 0) are accessed,
- a byte operation, the lower 8 bits (bits 7 to 0) are accessed, or
- a nibble operation, the lower 4 bits (bits 3 to 0) are accessed.

If the targeted bit in a bit instruction is specified by ACC (such as SBR, RBR, etc.), the upper 5 bits (bits 7 to 3) within the lower 8 bits specify the address offset, and the lower 3 bits (bits 2 to 0) specify the bit position.

ACC is assigned to the SFR area. At reset (due to a $\overline{\text{RES}}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of ACC become 0000H.

2.3.2 Control Registers

Control registers are a group of four 16-bit registers with dedicated functions for program status, program sequence, local registers and stack control.

(1) Program status word (PSW)

PSW is a 16-bit register consisting of the following.

- A flag (DD) that is referenced when executing instructions
- Flags (CY, ZF, HC, S, OV) that are set to "1" or reset to "0" depending upon instruction execution results
- Flags (SCB0 to SCB2) that specify the pointing register setting
- A flag (MIE) that enables ("1") or disables ("0") all maskable interrupts
- Flags (BCB0, BCB1) that specify the segment 0 common area
- Flags (F0 to F2) that the user can freely utilize
- A flag for use with future expanded CPU core functions. This flag (MAB) can be freely utilized by the user.

In addition to 16-bit PSW operations, 8-bit operations can also be performed with the PSW divided into the 8-bit units of PSWH (bits 15 to 8) and PSWL (bits 7 to 0).

Figure 2-8 shows the PSW configuration.

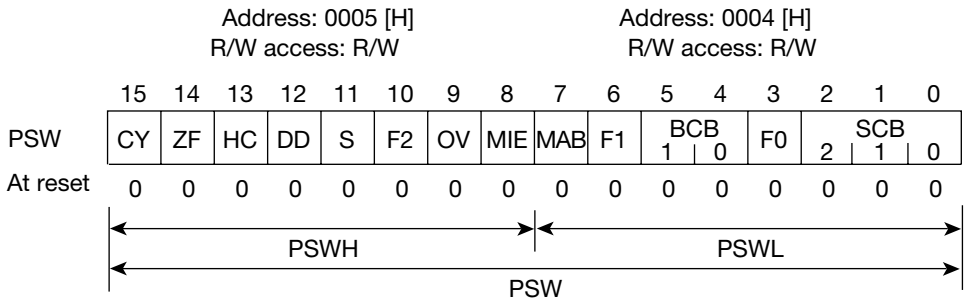


Figure 2-8 PSW Configuration

The upper 8 bits of the PSW (PSWH) contain:
- a flag (DD) that is referenced when executing instructions and
- flags (CY, ZF, HC, S, OV) that are set to "1" or reset to "0" depending upon instruction execution results.

Therefore, if the following instructions are performed on PSW or PSWH, flag operation may change from its original function.

(i) Instructions that load the contents of PSW or PSWH into ACC
(contents of ZF become undefined)
(ii) Bit operation instructions on ZF
(ZF changes depending on its value immediately before execution of the bit operation instruction.)
(iii) Increment, decrement, arithmetic, logic and compare instructions on PSW or PSWH
(The contents of PSW or PSWH immediately after instruction execution are undefined.)

If an interrupt occurs, PSW is automatically saved during interrupt processing and automatically restored by execution of a RTI instruction.

PSW is assigned to the SFR area. At reset (due to a $\overline{\text{RES}}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of PSW become 0000H.

Each bit in the PSW is described below.

Bit 15: Carry flag (CY)
The carry flag is set to "1" if:

- carry from bit 7 occurs in a byte operation,
- borrow to bit 7 occurs in a byte operation,
- carry from bit 15 occurs in a word operation, or
- borrow to bit 15 occurs in a word operation

as the result of executing an arithmetic or comparison instruction. Otherwise it is reset to "0". The carry flag can be set or reset directly by instructions and can be used to transmit or receive data for bits specified by registers. In addition, the carry flag can be tested by conditional branch instructions.

Bit 14: Zero flag (ZF)
The zero flag is set to "1" when:
- the result of an arithmetic instruction is zero,
- an instruction to load the ACC is executed and the load contents are zero, or
- a bit operation instruction is executed and the target bit is zero.

Otherwise, it is reset to "0". The zero flag can be tested by conditional branch instructions.

Bit 13: Half carry flag (HC)

The half carry flag is set to "1" if a carry or borrow from bit 3 occurs as a result of executing an arithmetic or comparison instruction (either a byte and word instruction). Otherwise, it is reset to "0".

Bit 12: Data descriptor (DD)

This flag indicates the attributes of data stored in ACC.

- When DD is "1", the 16 bits of data in ACC are determined to be valid.
- When DD is "0", the lower 8 bits of data in ACC are determined to be valid.

Instructions that reference DD when performing arithmetic or data transfer instructions with ACC are executed as follows.

- When DD is "1", the arithmetic or transfer operation is performed in word units.
- When DD is "0", the arithmetic or transfer operation is performed in byte units.

DD is set to "1" or reset to "0" when a data transfer instruction to ACC is executed and when dedicated set and reset instructions are executed.

- DD is set to "1" when executing a word-type load instruction to ACC and when executing a SDD instruction.
- DD is reset to "0" when executing a byte-type load instruction to ACC and when executing a RDD instruction.

If DD is modified (set or reset) while executing a load instruction to ACC or a dedicated set or reset instruction, and if the next instruction references DD, the modified DD will be referenced.

Since DD is assigned to PSW, DD can be overwritten by instructions other than those mentioned above. In this case, if the next instruction references DD, it will reference the state of DD prior to modification. If DD is to be used in this manner, insert a NOP instruction after the instruction that directly modifies the state of DD.

Bit 11: Sign flag (S)

The sign flag is set to "1" if the MSB of the result of executing an arithmetic or logic instruction is "1". If the MSB of the result is "0", the sign flag is reset to "0".

Bit 10: User flag 2 (F2)
Bit 6: User flag 1 (F1)
Bit 3: User flag 0 (F0)

These flags can be set to "1" or reset to "0" by instructions.

Bit 9: Overflow flag (OV)

The overflow flag is set to "1" if the result of executing an arithmetic instruction exceeds a range expressed in 2's compliment format (–128 to +127 for byte operations and –32,768 to +32,767 for word operations). Otherwise the overflow flag is reset to "0".

Bit 8: Master interrupt enable flag (MIE)

The master interrupt enable flag enables ("1") or disables ("0") all maskable interrupts.

During a maskable interrupt transfer cycle, after this flag is saved onto the system stack as part of PSW, it is reset to "0", and then restored by execution of a RTI instruction. If MIE is set to "1", the generation of all maskable interrupts is enabled from the next instruction. If reset to "0", the generation of all maskable interrupts is disabled from the next instruction.

Bit 7: Product-sum function bank flag (MAB)

The MSM66577 family does not have the product-sum function, so this can be utilized as a user flag.

Bit 5: Bank common base 1 (BCB1)
Bit 4: Bank common base 0 (BCB0)

These flags specify the last address of the common area between segments in data memory space. The table below shows the relation between BCB value and selected common area.

B C B		BCB range of common area
1	0	
0	0	0H to 03FFH
0	1	0H to 1FFFH
1	0	0H to 3FFFH
1	1	0H to 7FFFH

Bit 2: System control base 2 (SCB2)
Bit 1: System control base 1 (SCB1)
Bit 0: System control base 0 (SCB0)

These flags specify the pointing register (PR) set assigned to the fixed page area.

S C B			SCB pointing register set
2	1	0	
0	0	0	PR0(0200H to 0207H)
0	0	1	PR1(0208H to 020FH)
0	1	0	PR2(0210H to 0217H)
0	1	1	PR3(0218H to 021FH)
1	0	0	PR4(0220H to 0227H)
1	0	1	PR5(0228H to 022FH)
1	1	0	PR6(0230H to 0237H)
1	1	1	PR7(0238H to 023FH)

(2) Program counter (PC)

The PC is a 16-bit counter that stores the next address to be executed in the program segment. The PC is normally incremented according to the number of bytes in the instruction to be executed. If a branch instruction or an instruction that requires a branch is executed, the PC is loaded with immediate data, register contents, etc. The CSR value does not change even if the PC is incremented so that it overflows.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), or when an interrupt is generated, a value from the vector table is loaded into the PC.

(3) Local register base (LRB)

LRB is a 16-bit register. The lower 8 bits (LRBL) specify the 2KB data memory space from 0200H to 09FFH in 8-byte units (local register addressing). The upper 8 bits (LRBH) specify the 64KB data memory space in 256-byte units of the segment (segments 0 to 15) arbitrarily specified by the data segment register (DSR) (current page addressing). SB, RB, JBR and JBS instructions whose object is sba.bit can be used in the 64-byte area of the current page from xxC0H to xxFFH.

Both LRBL (02H) and LRBH (03H) are assigned to the SFR area. At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), their value is undefined.

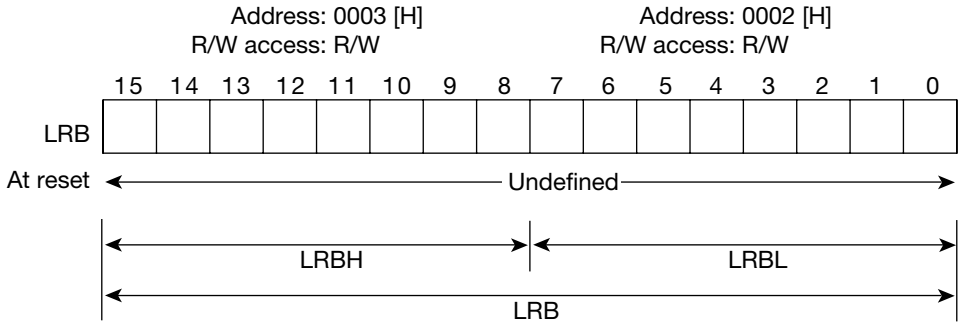
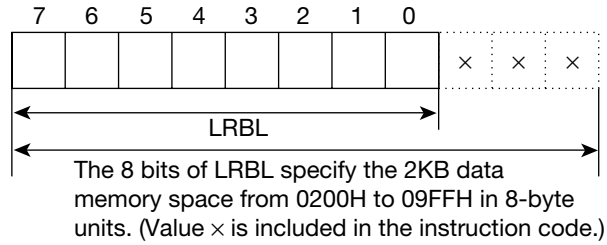
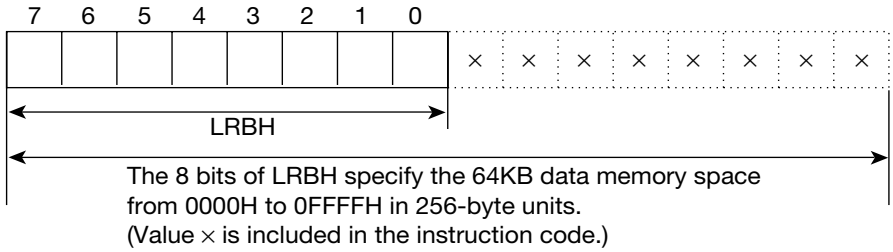


Figure 2-9 LRB Configuration

- The 8 bits of LRBL specify the 2KB data memory space from 0200H to 09FFH in 8-byte units.



- The 8 bits of LRBH specify 64KB of data memory space in 256-byte units.



(4) System stack pointer (SSP)

SSP is a 16-bit register that indicates the stack address at which to save or restore the PC, registers, etc. while processing interrupts or executing call, push, return, or pop instructions. SSP is automatically incremented or decremented depending upon the process to be executed.

Since save and restore operations at the address indicated by the SSP are performed in word units, the least significant bit (LSB) of the SSP is addressed as "0". The SFR area and the Expanded SFR area can not be used as a stack area.

SSP (00H) is assigned to the SFR area. At reset (due to a $\overline{\text{RES}}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of SSP become 0FFFFH.

2.3.3 Pointing Register (PR)

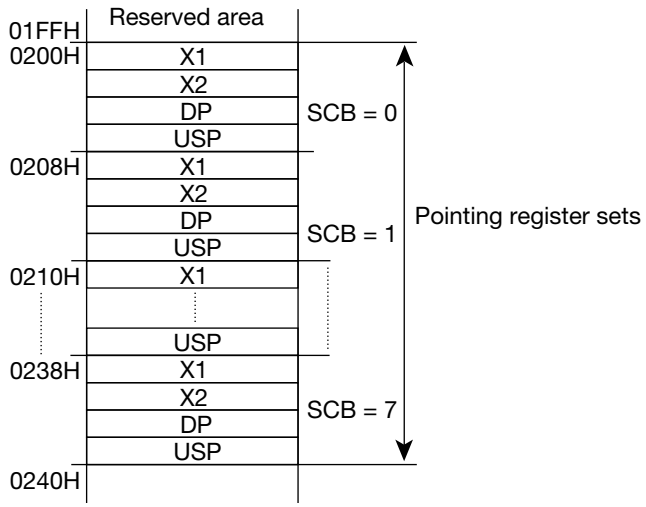
The PR has 8 sets of registers. One set consists of the following four 16-bit registers.

- Index register 1 (X1)
- Index register 2 (X2)
- Data pointer (DP)
- User stack pointer (USP)

PR is assigned to the internal RAM space from 0200H to 023FH. One of the eight register sets is selected by SCB0 to SCB2 of PSWL.

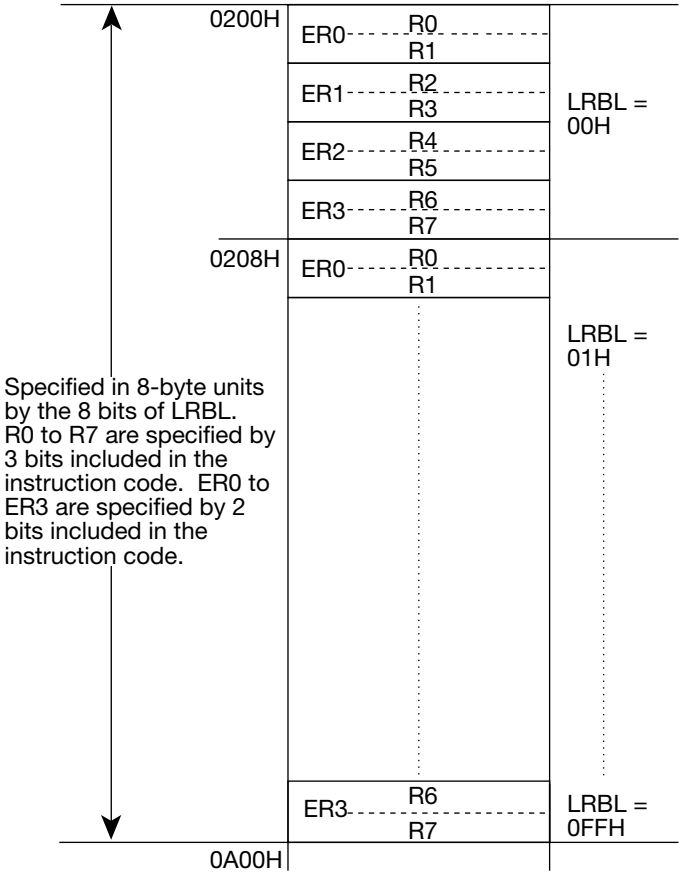
If the PR function is not used, this area can be used as normal internal RAM.

For all X1, X2, DP and USP, even addresses are the lower 8 bits and the following odd addresses are the upper 8 bits.



2.3.4 Local Registers (R0 to R7, ER0 to ER3)

The local register Rn (n = 0 to 7) is an 8-bit register and the expanded local register ERm (m = 0 to 3) is a 16-bit register. The 2KB area in data memory space from 0200H to 09FFH is specified in 8-byte units by the lower 8 bits of local register base (LRBL). Rn accesses 1 byte of the specified 8 bytes according to the 3 bits of data included in the local register instruction. (ERm accesses 2 bytes according to the 2 bits of data included in the local register instruction.)



2.3.5 Segment Registers

There are three 8-bit segment registers: the code segment register (CSR), the table segment register (TSR) and the data segment register (DSR). These registers select segments in the program memory space.

However, since the program memory space has only segments 0 to 15, only bits 0 to 3 are valid. Bits 4 to 7 are fixed to "0".

(1) Code segment register (CSR)

	7	6	5	4	3	2	1	0
CSR	"0"	"0"	"0"	"0"				
At reset	0	0	0	0	0	0	0	0

CSR specifies the segment in program memory space to which the program code currently being executed belongs. CSR exists as an independent 8-bit register and is not assigned to the SFR area. The CSR contents can be overwritten by FJ, FCAL, VCAL, FRT and RTI instructions and interrupts. No other methods can be used to overwrite the contents of CSR. For FJ and FCAL instructions, use branch destination addresses that are within segments 0 to 15.

Each segment is assigned an internal segment offset address of 0 to 0FFFFH. The address calculation to determine the addressed target is performed with a 16-bit offset address and any resulting overflow or underflow is ignored so that CSR does not change. Similarly, overflow of the PC never updates the CSR. Therefore, without the use of the CSR overwrite method described above, program execution does not advance beyond the code segment boundary. The CSR value at reset is 00H.

When an interrupt occurs after program memory space has been expanded to 1MB, both the current CSR value and the PC are automatically saved on the stack. Executing a RTI instruction restores the saved value to CSR. (Refer to Section 2.2.1, "Memory Space Expansion".)

(2) Table segment register (TSR)

	7	6	5	4	3	2	1	0
TSR	"0"	"0"	"0"	"0"				
At reset	0	0	0	0	0	0	0	0

Address: 0008 [H]
R/W access: R/W

"0" indicates that a value of "0" must be written.
If read, a value of "0" will be obtained.

TSR specifies the segment in program memory to which the table data belongs. TSR is an 8-bit register and is assigned to the SFR area. The contents of TSR can be overwritten by instructions that use SFR addressing. Data in the table segment can be accessed by using ROM reference instructions (LC, LCB, CMPC and CMPCB). If the ROM window function is used, RAM addressing can be utilized for this table segment. Only bits 0 to 3 of TSR are valid. If read, a value of "0" will be obtained for bits 4 to 7. If writing to TSR, "0" must be written to bits 4 to 7.

Each segment is assigned an internal segment offset address of 0 to 0FFFFH. The address calculation to determine the addressed target is performed with a 16-bit offset address and any resulting overflow or underflow is ignored, so TSR does not change. The TSR value at reset is 00H.

(3) Data segment register (DSR)

	7	6	5	4	3	2	1	0
DSR	"0"	"0"	"0"	"0"				
At reset	0	0	0	0	0	0	0	0

Address: 0009 [H]
R/W access: R/W

"0" indicates that a value of "0" must be written.
If read, a value of "0" will be obtained.

DSR specifies the segment in data memory space to which the data currently in use belongs. DSR is an 8-bit register and is assigned to the SFR area. The contents of DSR can be overwritten by instructions that use SFR addressing. Only bits 0 to 3 of DSR are valid. If read, a value of "0" will be obtained for bits 4 to 7. If writing to DSR, "0" must be written to bits 4 to 7.

2.4 Addressing Modes

The MSM66577 family has two independent memory spaces, the data memory space and the program memory space. Addressing can be roughly classified into two modes, corresponding to each memory space.

The data memory space is referred to as "RAM space", since it normally consists of random access memory (RAM). The addressing for this space is referred to as "RAM addressing".

The program memory space is referred to as "ROM space", since it normally consists of read-only memory (ROM). The addressing for this space is referred to as "ROM addressing".

ROM addressing is classified as immediate addressing contained in instruction codes, table data addressing for data (normally read-only data) in a ROM space table, and program code addressing for programs in the ROM space.

ROM window addressing is a unique method of addressing. It involves accessing table data in the ROM space using the above RAM addressing methods. Data in a table segment is read through a data segment window specified and opened by the program.

2.4.1 RAM Addressing

This addressing mode specifies addresses for program variables in the RAM space.

Available addressing formats include: register addressing, page addressing, direct addressing, pointing register indirect addressing and special bit area addressing.

(1) Register addressing

A.	Accumulator addressing	A
B.	Control register addressing	PSW, LRB, SSP
C.	Pointing register addressing	X1, X2, DP, USP
D.	Local register addressing	ERn, Rn

A. Accumulator addressing

In the case of a word-format instruction, the contents of the accumulator (A) will be accessed. In the case of byte and bit-format instructions, the lower byte of the accumulator (AL) will be accessed.

[Word format]

```
L       A, #1234H
ST      A, VAR
```

[Byte format]

```
LB      A, #12H
STB     A, VAR
```

[Bit format]

```
MB      C, A.3
JBS     A.3, LABEL
```

B. Control register addressing

The contents of the registers will be accessed.

SSP:	System Stack Pointer
LRB:	Local Register Base
PSW:	Program Status Word
PSWH:	Program Status Word High Byte
PSWL:	Program Status Word Low Byte
C:	Carry Flag

[Word format]

```
FILL    SSP
MOV     LRB, #401H
CLR     PSW
```

[Byte format]

```
CLRB    PSWH
INCB    PSWL
```

[Bit format]

```
MB      C, BITVAR
```

C. Pointing register addressing

The contents of the pointing register are accessed.
There are 8 sets of pointing registers (PR0 to PR7: every 8 bytes from 200H to 23FH in data memory). The set addressed by this mode is specified by the value of the system control base (SCB) field in PSW.

X1:	Index Register 1
X2:	Index Register 2
DP:	Data Pointer The low byte of the data pointer is used only for a "JRNZ DP radr" instruction (to maintain compatibility among nX-8/100 to nX-8/400 CPU cores).
USP:	User Stack Pointer
X1L:	Index Register 1 Low Byte
X2L:	Index Register 2 Low Byte
DPL:	Data Pointer Low Byte
USPL:	User Stack Pointer Low Byte

[Word format]

```
L     A, X1
ST    A, X2
MOV   DP, #2000H
CLR   USP
```

[Byte format]

```
DJNZ  X1L, LOOP
DJNZ  X2L, LOOP
DJNZ  DPL, LOOP
DJNZ  USPL, LOOP
JRNZ  DP, LOOP
```

D. Local register addressing

The contents of the local register are accessed.

There are 256 sets of local registers (every 8 bytes from 200H to 9FFH in data memory). The set addressed by this mode is specified by the value of the low byte of the local register base (LRB).

ER0 to ER3:	Expanded Local Registers
R0 to R7:	Local Registers

[Word format]

```
L       A, ER0
MOV     ER2, ER1
CLR     ER3
```

[Byte format]

```
LB      A, R0
ADDB    R1, A
CMPB    R2, #12H
INCB    R3
ROR     R4
MOVB    R5, R6
```

[Bit format]

```
SB      R0.0
RB      R1.7
JBRS    R7.3, LABEL
```

(2) Page addressing

A.	SFR page addressing	sfr Dadr
B.	FIXED page addressing	fix Dadr
C.	Current page addressing	off Dadr

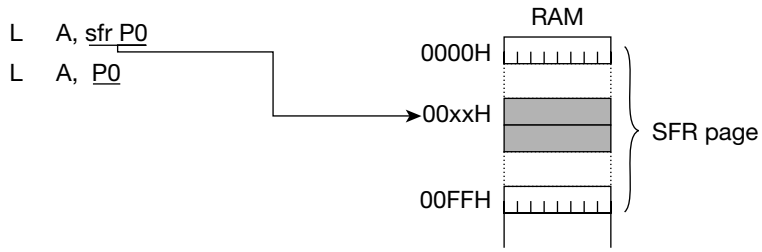
A. SFR page addressing

One byte of the instruction code specifies an offset within a SFR page (data memory addresses 0 to 0FFH). Word-format, byte-format or bit-format data at the specified address is accessed.

The operand is described using a format that has a sfr addressing descriptor. The sfr descriptor can be omitted, however in that case, the assembler will use SFR page addressing only when it recognizes an address within the SFR page area.

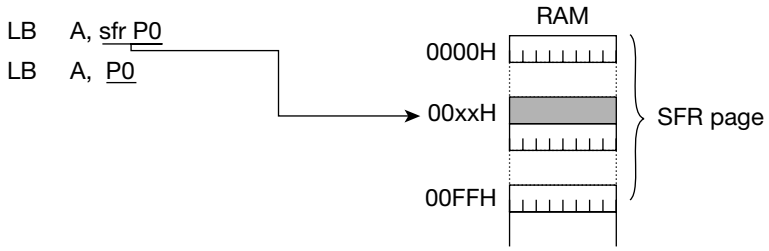
The SFR has address symbols for each type of device. These symbols are normally used for addressing the SFR.

[Word format]

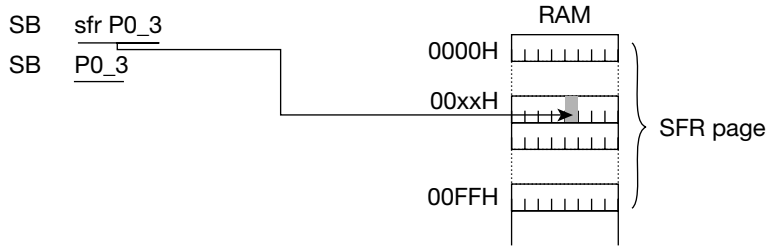


If an odd address is specified, word-format data is accessed starting at the following even address. (→word boundary) However, depending upon the SFR, there are some exceptions.

[Byte format]



[Bit format]

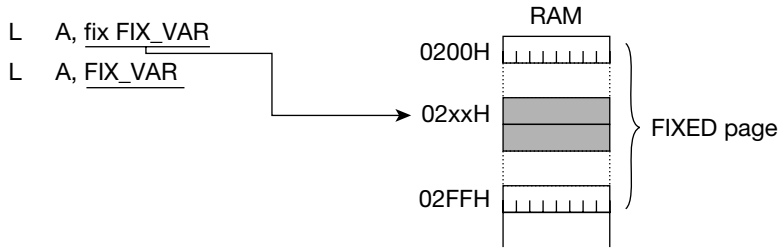


B. FIXED page addressing

One byte of the instruction code specifies an offset within a FIXED page (data memory addresses 200H to 2FFH). Word-format, byte-format or bit-format data at the specified address is accessed.

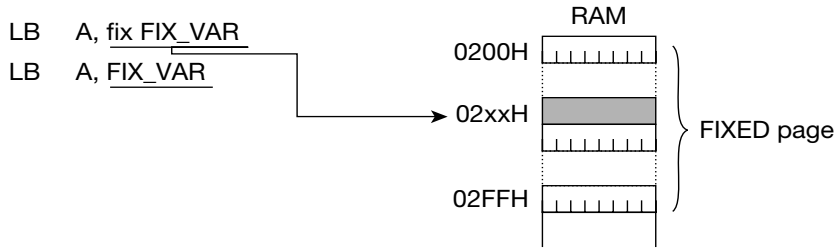
The operand is described using a format that has a fix addressing descriptor. The fix descriptor can be omitted, however in that case, the assembler will use FIXED page addressing only when it recognizes an address within the FIXED page area.

[Word format]

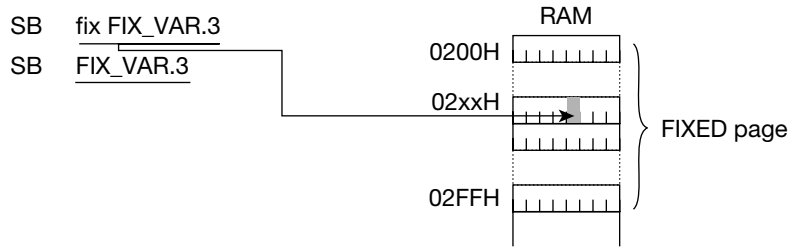


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

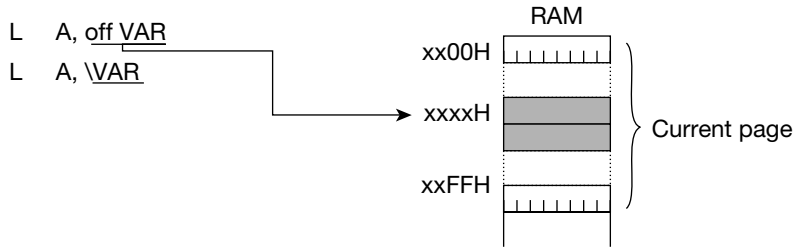


C. Current page addressing

One byte of the instruction code specifies an offset within the current page (one of the 256 pages in data memory specified by the LRBH value). Word-format, byte-format or bit-format data at the specified address is accessed.

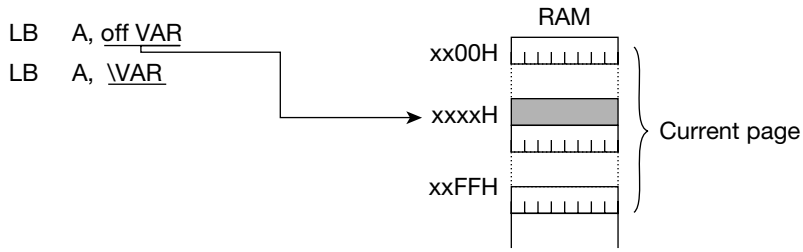
The operand is described using a format that has an off addressing descriptor. \ can be used instead of the off descriptor, however if bit-format data is accessed in the SBA area, operation will be slightly different. (sbaoff Badr)

[Word format]

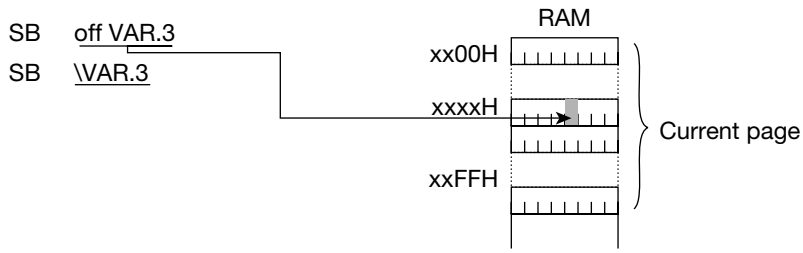


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

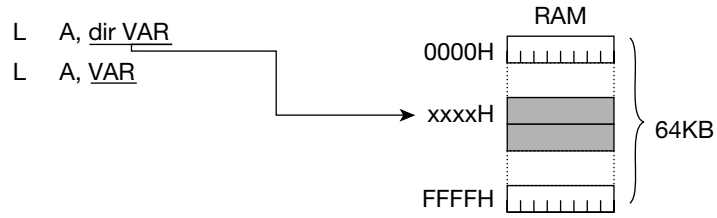


(3) Direct data addressing

Two bytes of the instruction code specify an address in the current physical segment of data memory (address 0 to 0FFFFH: 64KB). Word-format, byte-format or bit-format data at the specified address is accessed.

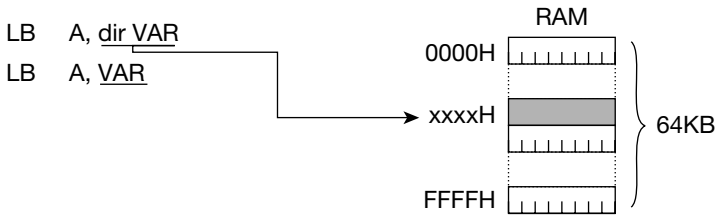
The operand is described using a format that has a dir addressing descriptor. The dir descriptor can be omitted, however in this case, if an address in a SFR page or FIXED page is specified, the assembler may interpret direct data addressing as SFR page addressing or FIXED page addressing.

[Word format]

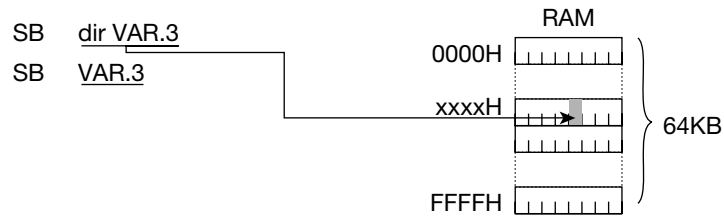


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



(4) Pointing register indirect addressing

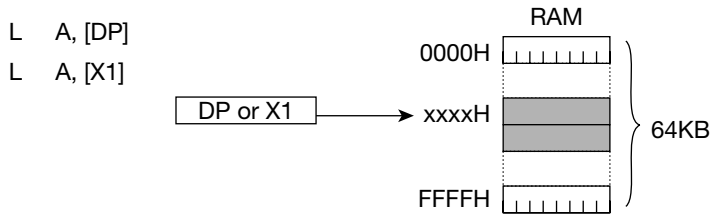
A.	DP/X1 indirect addressing	[DP], [X1]
B.	DP indirect addressing with post increment	[DP+]
C.	DP indirect addressing with post decrement	[DP-]
D.	DP/USP indirect addressing with 7-bit displacement	n7[DP], n7[USP]
E.	X1/X2 indirect addressing with 16-bit base	D16[X1], D16[X2]
F.	X1 indirect addressing with 8-bit register displacement	[X1+R0], [X1+A]

A. DP/X1 indirect addressing

The contents of the pointing register specify an address in the current physical segment of data memory (address 0 to 0FFFFH: 64KB). Word-format, byte-format or bit-format data at the specified address is accessed.

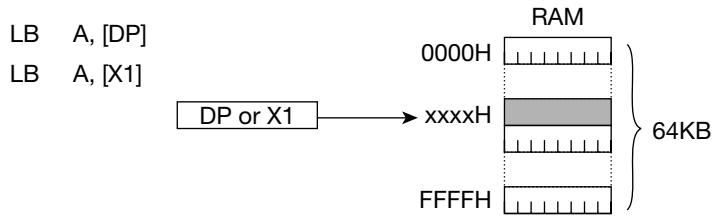
[DP]: DP indirect addressing
[X1]: X1 indirect addressing

[Word format]

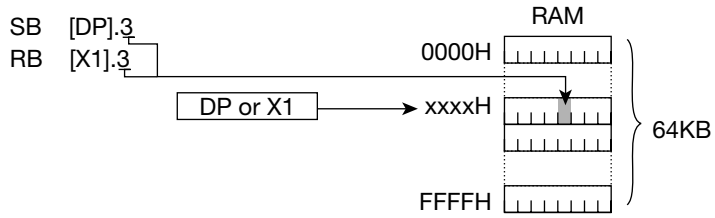


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



B. DP indirect addressing with post increment

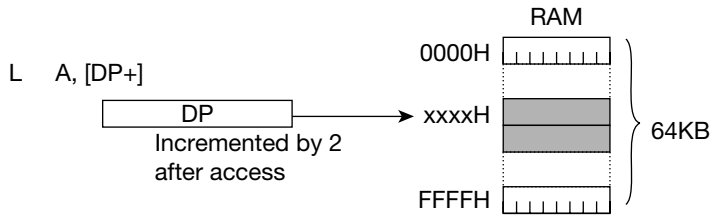
The contents of the pointing register specify an address in the current physical segment of data memory (address 0 to 0FFFFH: 64KB). Word-format, byte-format or bit-format data at the specified address is accessed.

After accessing the target, the contents of the pointing register are incremented. For word-format instructions, DP is incremented by two. For byte and bit instructions, DP is incremented by one.

This addressing mode is used primarily to consecutively access an array of elements.

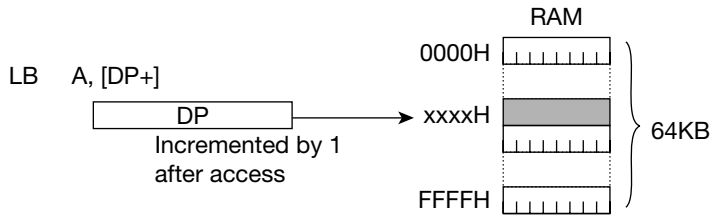
[DP+]: DP indirect addressing with post increment

[Word format]

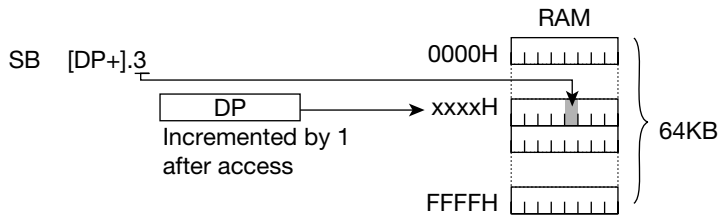


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



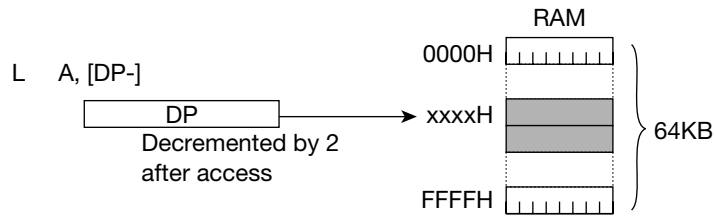
C. DP indirect addressing with post decrement

The contents of the pointing register specify an address in the current physical segment of data memory (addresses 0 to 0FFFFH: 64KB). Word-format, byte-format or bit-format data at the specified address is accessed.

After accessing the target, the contents of the pointing register are decremented. For word-format instructions, DP is decremented by two. For byte and bit instructions, DP is decremented by one.

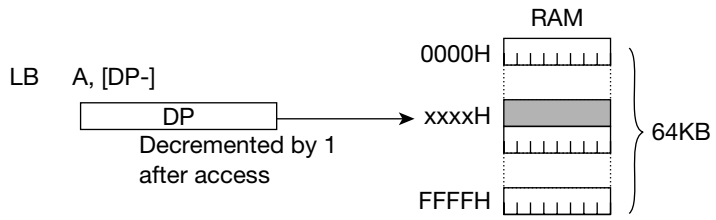
[DP-]: DP indirect addressing with post decrement

[Word format]



If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



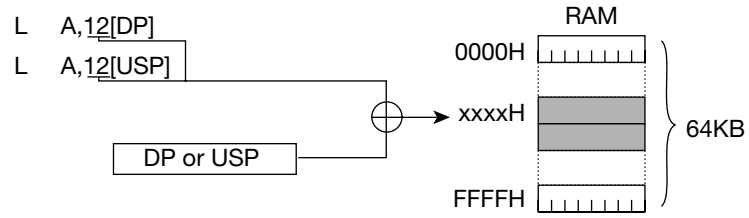
[Bit format]

D. DP/USP indirect addressing with 7-bit displacement

7 bits in the instruction code (bit 6 to bit 0) are used as a signed displacement (bit 6 is the sign bit) from the pointing register contents (the base value) to specify an address in the current physical data segment (address 0 to 0FFFFH: 64KB). The accessible range is –64 to +63 from the contents of the pointing register. Word-format, byte-format or bit-format data at the specified address is accessed.

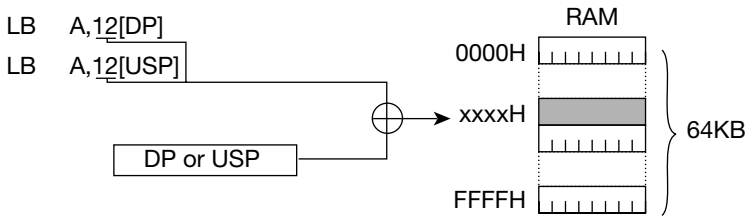
Numerical expression[DP]: DP indirect addressing with 7-bit displacement
Numerical expression[USP]: USP indirect addressing with 7-bit displacement
The numerical expression has a value in the range of –64 to +63.
DP and USP can be used as pointing registers.

[Word format]

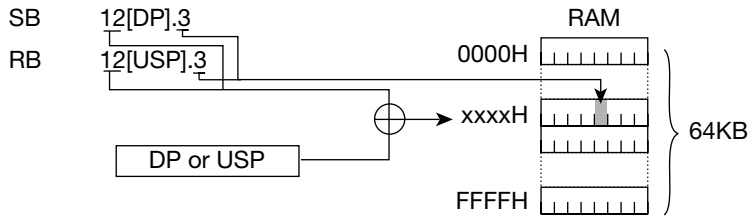


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

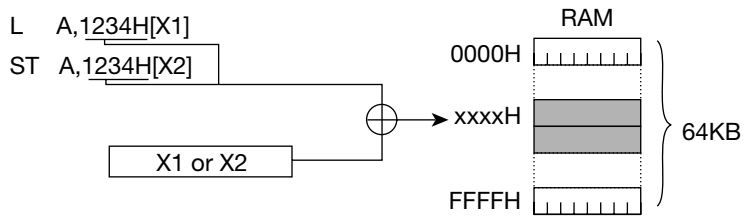


E. X1/X2 indirect addressing with 16-bit base

The contents of an index register (X1 or X2) are added to a base of two bytes in the instruction code (D16). The value that is generated specifies an address in the current physical data segment (address 0 to 0FFFFH: 64KB). The addition operation to generate the address is performed in word-format (16-bit) and since overflow is ignored, the generated value is in the range from 0 to 0FFFFH. Word-format, byte-format or bit-format data at the specified address is accessed.

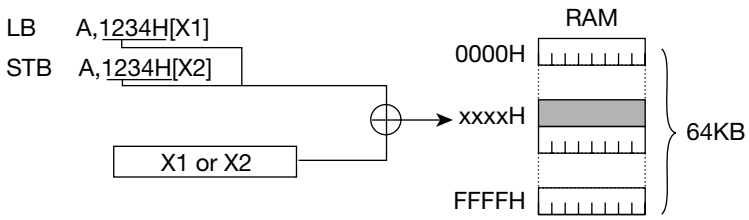
Address expression[X1]: X1 indirect addressing with 16-bit base
Address expression[X2]: X2 indirect addressing with 16-bit base
The address expression has a value in the range of 0 to 0FFFFH. However, the assembler allows values in the range of –8000H to +0FFFFH. This means that D16 can also be regarded as a displacement, instead of a base address.

[Word format]

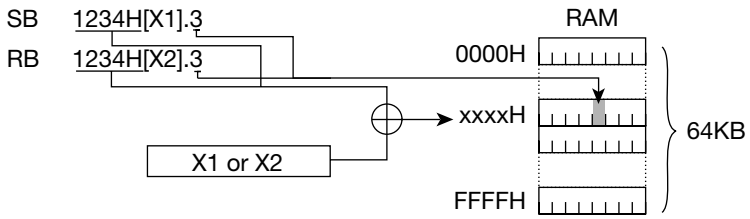


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



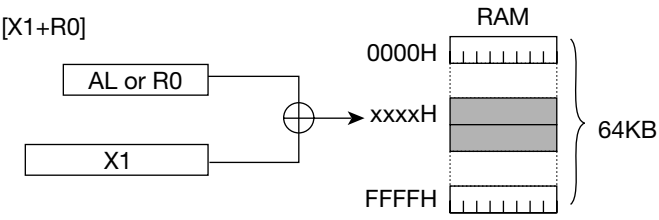
F. X1 indirect addressing with 8-bit register displacement

The contents of the low byte of the accumulator (AL) or local register 0 (R0) are added to the pointing register contents (the base value) to generate a value that specifies an address in the current physical data segment (address 0 to 0FFFFH: 64KB). The addition operation to generate the address is performed in word-format (16-bit). At this time, the 8-bit displacement obtained from the register is expanded unsigned. Since overflow resulting from the addition is ignored, the generated value is in the range from 0 to 0FFFFH. Word-format, byte-format or bit-format data at the specified address is accessed.

[X1+A]: X1 indirect addressing with 8-bit register displacement (AL)
[X1+R0]: X1 indirect addressing with 8-bit register displacement (R0)

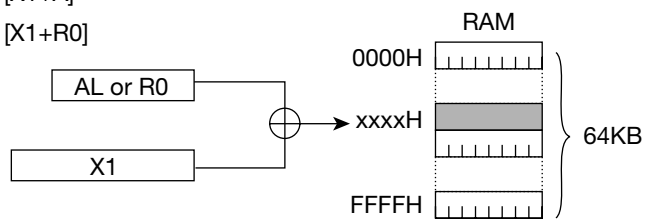
[Word format]

```
L   A, [X1+A]
L   A, [X1+R0]
```



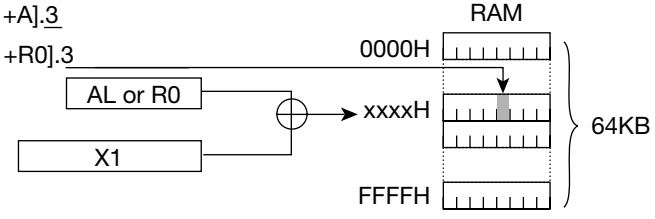
If an odd address is specified, word-format data is accessed starting at the following even address.

```
LB  A, [X1+A]
LB  A, [X1+R0]
```



[Byte format]

```
SB  [X1+A].3
RB  [X1+R0].3
```



[Bit format]

(5) Special bit area addressing

A. FIXED page SBA area addressing — sbafix Badr
B. Current page SBA area addressing — sbaoff Badr

A. FIXED page SBA area addressing

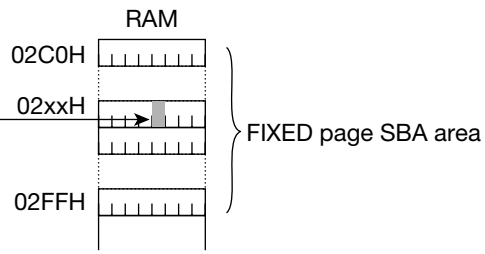
This addressing mode specifies a bit address in the 512-bit SBA area (2C0H.0 to 2FFH.7) located in a FIXED page. Bit format data at the specified address is accessed.

This addressing mode can be written by the following 4 instructions: SB, RB, JBS and JBR.

[Bit format]

```
SB   sbafix 2C0H.0
RB   sbafix 1600H
JBS  sbafix VAR,LABEL
JBR  sbafix 2EFH.7
```

```
SB   2C0H.0
RB   1600H
JBS  VAR,LABEL
JBR  2EFH.3,LABEL
```



B. Current page SBA area addressing

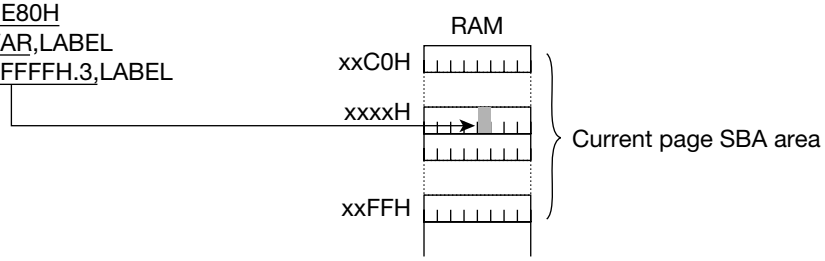
This addressing mode specifies a bit address in the 512-bit SBA area (xxC0H.0 to xxFFH.7) located in the current page. Bit format data at the specified address is accessed.

This addressing mode can be written by the following 4 instructions: SB, RB, JBS and JBR.

[Bit format]

```
SB    sbaoff 4C0H.0
RB    sbaoff 2E80H
JBS   sbaoff VAR,LABEL
JBR   sbaoff 0FFFFH.3,LABEL
```

```
SB    2C0H.0
RB    2E80H
JBS   VAR,LABEL
JBR   0FFFFH.3,LABEL
```



2.4.2 ROM Addressing

This addressing mode specifies addressing for program variables in the ROM space.

Available addressing formats include: immediate addressing, table data addressing and program code addressing.

(1) Immediate addressing

This addressing mode specifies access for immediate data included in the instruction code. For word-format instructions, 2 bytes (N16) of the instruction code are accessed. For byte-format instructions, 1 byte (N8) of the instruction code is accessed.

In the word-format, expressions have values in the range of 0 to 0FFFFH. In the byte-format, expressions have values in the range of 0 to 0FFH. The assembler allows a range of signed and unsigned expressions for immediate addressing. The word-format range is from -8000H to +0FFFFH and the byte-format range is from -80H to +0FFH.

[Word format]

```
L       A, #1234H
MOV     X1, #WORD_ARRAY_BASE
```

[Byte format]

```
LB      A, #12H
MOV     X1, #BYTE_ARRAY_BASE
```

(2) Table data addressing

This addressing mode specifies access for 64KB in the table segment specified by TSR in ROM memory space. This mode is used with the operands of LC, LCB, CMPC and CMPCB instructions.

A.	Direct table addressing	Tadr
B.	RAM addressing indirect table addressing	[**]
C.	RAM addressing indirect addressing with 16-bit base	T16[**]

A. Direct table addressing

Two bytes of the instruction code specify an address (address 0 to 0FFFFH: 64KB) in the table segment specified by TSR. Word-format or byte-format data at the specified address is accessed.

This addressing mode can be written by the following 4 instructions: LC, LCB, CMP and CMPCB.

[Word format]
```
LC      A, VAR
CMPC    A, VAR
```

[Byte format]
```
LCB     A, VAR
CMPCB   A, VAR
```

B. RAM addressing indirect table addressing

This indirect addressing mode uses the word-format data specified by RAM addressing as a pointer to the table segment specified by TSR. Table memory can be accessed by placing a pointer to table memory in a register or in data memory.

This addressing mode can be written by the following 4 instructions: LC, LCB, CMPC and CMPCB.

[Word format]
```
LC      A, [A]
CMPC    A, [1234[X1]]
```

[Byte format]
```
LCB     A, [ER0]
CMPCB   A, [VAR]
```

C. RAM addressing indirect addressing with 16-bit base

The contents of word-format data specified by RAM addressing are added to a base of two bytes of the instruction code (D16). The value that is generated specifies an address in the table segment specified by TSR (address 0 to 0FFFFH: 64KB). The addition operation to generate the address is performed in word-format (16-bit) and since overflow is ignored, the generated value is in the range from 0 to 0FFFFH. Word-format or byte-format data at the specified address is accessed.

This addressing mode can be written by the following 4 instructions: LC, LCB, CMPC and CMPCB.

[Word format]
```
LC      A, 2000H[A]
CMPC    A, 2000H[1234[X1]]
```

[Byte format]
```
LCB     A, 2000H[ER0]
CMPCB   A, 2000H[VAR]
```

(3) Program code addressing

This mode specifies access for the current program code in ROM space.

Program code addressing is used with operands for branch instructions.

A. NEAR code addressing	Cadr
B. FAR code addressing	Fadr
C. Relative code addressing	radr
D. ACAL code addressing	Cadr11
E. VCAL code addressing	Vadr
F. RAM addressing indirect code addressing	[**]

A. NEAR code addressing

Two bytes of the instruction code specify an address (address 0 to 0FFFFH: 64KB) in the current code segment.

This addressing mode can be written by two instructions, J and CAL.

[Usage example]

```
J       3000H
CAL     LABEL
```

B. FAR code addressing

Three bytes of the instruction code specify an address (0:0 to F:0FFFFH: 1MB) in the program memory space.

This addressing mode can be written by two instructions, FJ and FCAL.

[Usage example]

```
FJ      1:3000H
FCAL    FARLABEL
```

C. Relative code addressing

The sign extended value of 8 bits or 7 bits of the instruction code is added to the base value of the current program counter (PC). The generated value specifies an address in the current code segment (0 to 0FFFFH: 64KB). The addition operation to generate the address is performed in word-format (16-bit) and since overflow is ignored, the generated value is in the range from 0 to 0FFFFH. This addressing mode can be written by an SJ instruction, conditional branch instructions, etc.

[Usage example]

```
SJ      LABEL
DJNZ    R0, LABEL
JC      LT, LABEL
```

D. ACAL code addressing

11 bits of the instruction code specify the ACAL area (1000H to 17FFH: 2KB) in the current code segment.

This addressing mode can be written only by an ACAL instruction.

[Usage example]
```
ACAL    1000H
ACAL    ACALLABEL
```

E. VCAL code addressing

4 bits of the instruction code specify the vector table address for a VCAL instruction (word-format data). The vector table is located at even addresses in the range of 004AH to 0069H.

This addressing mode can be written only by a VCAL instruction.

[Usage example]
```
VCAL    4AH
VCAL    0:4AH
VCAL    VECTOR
```

F. RAM addressing indirect code addressing

This indirect addressing mode uses the word-format data specified by RAM addressing as a pointer to the code segment. Indirect jumps and calls can be performed by placing a pointer to code memory in a register or in data memory.

This addressing mode can be written by two instructions, J and CAL.

[Usage example]
```
J       [A]
CAL     [1234[X1]]
```

(4) ROM window addressing

This addressing mode uses RAM addressing to access table data in the ROM space. In this mode, data in the table segment specified by TSR is read through a data segment window specified and opened by the program.

The ROM window area allows addressing of the data memory, however, results cannot be guaranteed if an instruction that writes to the ROM window area is executed.

Chapter 3

CPU Control Functions

3. CPU Control Functions

3.1 Overview

The MSM66577 family has two CPU control functions, a standby function and a reset function.

The standby function consists of the three functions of HALT mode, HOLD mode, and STOP mode. These functions can be used to reduce the amount of power consumed during operation. The HOLD mode has a bus release function, and the STOP mode has a quick activating STOP mode in which the main clock continues oscillation.

The reset function is activated by the $\overline{\text{RES}}$ signal input, BRK (break) instruction execution, or execution of an invalid instruction (opcode trap). In addition, reset is also activated by overflow of the watchdog timer. Reset can minimize the effect of program errors on the system.

3.2 Standby Functions

The standby functions have three types:

- HALT mode: activated by software, clock supply to CPU is terminated
- HOLD mode: activated by hardware, clock supply to CPU is terminated
- STOP mode: activated by software, clock supply to CPU and internal peripheral modules is terminated

Corresponding to each of dual clocks, each of these functions has a high-speed and low-speed mode.

Figure 3-1 shows a transition diagram of the CPU operating states. Table 3-1 lists a summary of the standby modes.

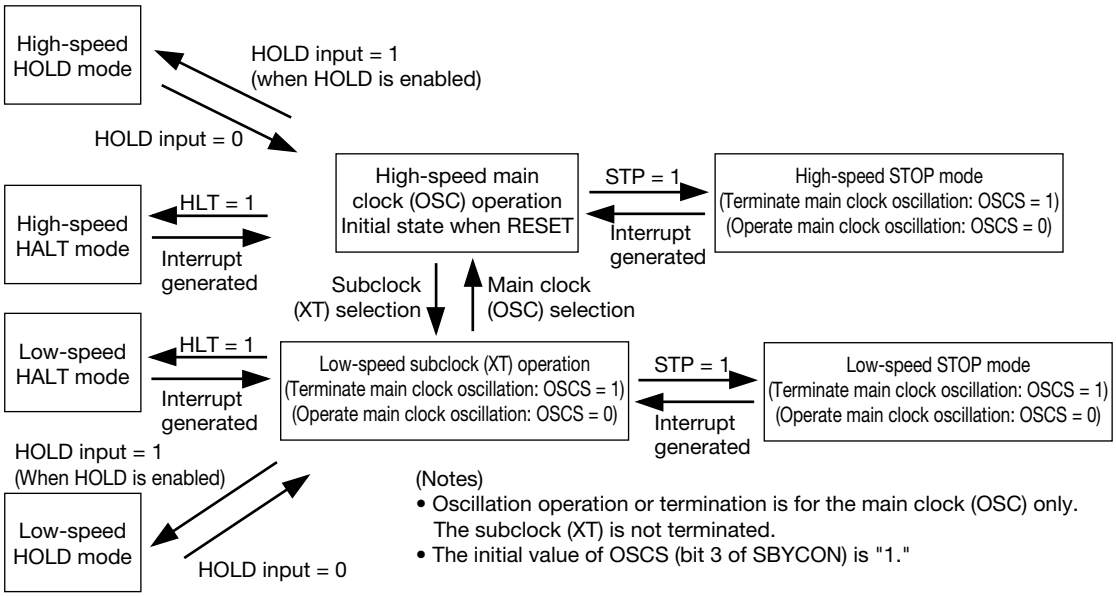


Figure 3-1 Transition Diagram of CPU Operating States

Table 3-1 Standby Mode Summary

Standby mode		HALT mode	HOLD mode	STOP mode *1	
Set conditions		Bit 1 (HLT) of SBYCON is set to "1"	Bit 5 (HOLD) of PRPHCON is set to "1" and the HOLD input pin has a high-level input	Bit 2 (FLT) of SBYCON is set to "1" and bit 0 (STP) of SBYCON is set to "1"	Bit 2 (FLT) of SBYCON is reset to "0" and bit 0 (STP) of SBYCON is set to "1"
Release conditions		Interrupt $\overline{RES}$ pin input WDT	HOLD pin low-level input $\overline{RES}$ pin input	Interrupt $\overline{RES}$ pin input	Interrupt $\overline{RES}$ pin input
Output pin states	P0 to P2, P4 (primary function)	No change	No change	High impedance	No change
	P0 to P2, P4 (secondary function)	Pull-up	Pull-up *2	Pull-up	Pull-up
	P3_0 (primary function)	No change	No change	High impedance	No change
	P3_0 (secondary function)	Low level	Low level	High impedance	Low level
	P3_1 (primary function)	No change	No change	High impedance	No change
	P3_1 (secondary function)	High level	Pull-up *2	High impedance	High level
	P3_2, P3_3 (primary function)	No change	No change	High impedance	No change
	P3_2, P3_3 (secondary function)	Pull-up	Pull-up *2	High impedance	Pull-up
	P5 to P11	No change	No change	High impedance	No change
	P14, P15	No change	No change	High impedance	No change
Operation of internal functions	Time base counter (TBC)	Operate	Operate	Terminate	
	Capture/compare timer	Operate	Operate	Terminate	
	8/16-bit timers (including WDT)	Operate	Operate (terminate WDT)	Terminate	
	SIO1, SIO6	Operate	Operate	Terminate	
	Real-time counter	Operate	Operate	Operate	
	A/D converter	Operate	Operate	Terminate	
	PWM	Operate	Operate	Terminate	

*1 The condition for setting the STOP mode is that the stop code acceptor (STPACP) has already been set to "1".

*2 During the HOLD mode, if P0 to P4 are to be used as bus ports (output setting of secondary function), the bus will be released.

3.2.1 Standby Function Registers

Table 3-2 lists a summary of the SFRs for standby function control.

Table 3-2 Summary of SFRs for Standby Function Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
000E	Stop code acceptor	STPACP	—	W	8	"0"	3-3
000F	Standby control register	SBYCON	—	R/W	8	08	3-4
0015☆	Peripheral control register	PRPHCON	—	R/W	8	8C	15-2

[Notes]

1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

3.2.2 Description of Standby Function Registers

(1) Stop code acceptor (STPACP)

The stop code acceptor (STPACP) is configured from 8 bits and is an acceptor used to set the STOP mode.

STPACP is set to "1" when the program writes n5H and nAH (n = 0 to F) consecutively. After STPACP is set to "1", setting bit 0 (STP) of the standby control register (SBYCON) to "1" will change the mode to the STOP mode. At the same time the mode changes to the STOP mode, STPACP is reset to "0".

STPACP is write-only.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), STPACP is reset to "0".

(2) Standby control register (SBYCON)

The standby control register (SBYCON) is an 8-bit register that sets the standby mode and the CPU operating clock (CPUCLK).

The program can read from and write to SBYCON.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), SBYCON is 08H.

Figure 3-2 shows the configuration of SBYCON.

[Description of each bit]

- STP (bit 0)
  Setting the stop code acceptor (STPACP) to "1", and then setting STP to "1" will change the mode to the STOP mode. When an interrupt is generated or the $\overline{RES}$ input causes a reset, STP is reset to "0" and the STOP mode is released.

- HLT (bit 1)
  Setting HLT to "1" changes the mode to the HALT mode. When an interrupt is generated, the $\overline{RES}$ input causes a reset, or overflow of the watchdog timer causes a reset, HLT is reset to "0" and the HALT mode is released.

- FLT (bit 2)
  Setting FLT to "1" will cause the output ports (all pins set to output mode) to go to a high impedance state when the STOP mode is entered.

  At the input ports, a circuit operates to prevent current flow between the power supply and GND, even if the inputs are left unconnected. Therefore, it is not necessary to fix the input pin levels during the STOP mode. However, if the following pins are used as inputs (regardless of whether they are primary or secondary functions), the circuit to prevent current flow will not operate. Thus, to prevent undefined input states, use either pull-up or pull-down resistors (to fix the input levels) during the STOP mode.

  - P6_0 to P6_3, P9_0 to P9_3: External interrupt pins (EXINT0 to EXINT7)

  Using the above pins as secondary function inputs, even if the STOP mode is entered with FLT set ("1"), the STOP mode can be released by an external interrupt input. For details, refer to Section 3.2.4, "Operation of Each Standby Mode," (3) STOP Mode.

- OSCS (bit 3)
  During the STOP mode and when the subclock (XTCLK) has been selected as the CPU operating clock (CPUCLK), OSCS specifies whether to terminate or continue oscillation of the main clock (OSCCLK).

- OST0, OST1 (bits 4 and 5)
  In the cases when an interrupt causes the STOP mode to be released, and when the clock has been changed from the subclock (XTCLK) to the main clock (OSCCLK), OST0 and OST1 specify the oscillation stabilization time from the oscillation start of the main clock (OSCCLK) until clock supply to the CPU. During the STOP mode, even if oscillation of the main clock (OSCCLK) is not terminated, the settings of these bits are valid.

  [Note]
  Do not set OST0 or OST1 to "1", in the case of changing to the operation mode in which oscillation of the main clock (OSCCLK) is terminated.
  For the Flash ROM version, set the oscillation stabilization time of 50 μs or more when the STOP mode (only when oscillation of the main clock is terminated) is released.

- CLK0, CLK1 (bits 6 and 7)
  CLK0 and CLK1 specify the clock to be used as the CPU operating clock (CPUCLK). With consideration of the operating speed requirements of product applications, an appropriate speed for the internal CPU clock that runs the microcontroller is selected to reduce power consumption.

	7	6	5	4	3	2	1	0
SBYCON	CLK1	CLK0	OST1	OST0	OSCS	FLT	HLT	STP
at reset	0	0	0	0	1	0	0	0

Address: 000F [H]
R/W access: R/W

0	CPU operating state
1	STOP mode

0	CPU operating state
1	HALT mode

0	During STOP mode, output does not change
1	During STOP mode, high impedance output

0	Main clock (OSC) oscillation
1	Terminate main clock (OSC) oscillation

OST		Main clock oscillation
1	0	stabilization time (No. of clocks)
0	0	32768
0	1	16384
1	0	8192
1	1	0

CLK		CPU operating clock
1	0	(CPUCLK)
0	0	OSCCLK
0	1	1/2 OSCCLK
1	0	1/4 OSCCLK
1	1	XTCLK

Figure 3-2 SBYCON Configuration

3.2.3 Examples of Standby Function Register Settings

• HALT mode setting

(1) Standby control register (SBYCON)
Setting bit 1 (HLT) to "1" changes the mode to the HALT mode.

• HOLD mode setting

(1) Port 9 mode register (P9IO)
If HLDACK (HOLD mode transfer output pin) is to be used, set bit 7 (P9IO7) to "1" to configure that port as an output.

(2) Port 9 secondary function control register (P9SF)
If HLDACK (HOLD mode transfer output pin) is to be used, set bit 7 (P9SF7) to "1" to configure that port as a secondary function output.

(3) Peripheral control register (PRPHCON)
Set bit 5 (HOLD) to "1" to enable the HOLD pin input. The mode changes to the HOLD mode when an external device inputs a high level to the HOLD pin. After the transfer to the HOLD mode is complete, the HLDACK output is set to "1" as an acknowledge signal.

• STOP mode setting

(1) Stop code acceptor (STPACP)
Write n5H, nAH (n = 0 to F) consecutively.

(2) Standby control register (SBYCON)
If output ports are to be high impedance during the STOP mode, set bit 2 (FLT) to "1". If oscillation of the main clock (OSCCLK) is not to be terminated during the STOP mode, reset bit 3 (OSCS) to "0". To terminate oscillation of the main clock (OSCCLK), set bit 3 (OSCS) to "1" and specify with bits 4 and 5 (OST0 and OST1) the oscillation stabilization time after the main clock resumes. Setting bit 0 (STP) to "1" changes the mode to the STOP mode.

3.2.4 Operation of Each Standby Mode

(1) HALT mode

Setting bit 1 (HLT) of the standby control register (SBYCON) to "1" changes the mode to the HALT mode.

In the HALT mode, the clock (CPUCLK) supply to the CPU is terminated, but the clock (CPUCLK) is supplied to internal peripheral modules (TBC, WDT, general-purpose 8/16-bit timers, serial ports, etc.) so their operation continues. Because the CPU is halted, instructions are not executed. Instruction execution stops at the beginning of the next instruction (following the instruction that set bit 1 (HLT) of SBYCON to "1").

HALT mode is released when any of the following occur: an interrupt request, reset by the $\overline{\text{RES}}$ pin input, or reset by overflow of the watchdog timer.

When HALT mode is released due to an interrupt request, if the interrupt is non-maskable, the HALT mode is released unconditionally, and the CPU processes the non-maskable interrupt. In the case of a maskable interrupt, the interrupt is released when both the interrupt request flag (IRQ bit) and the interrupt enable flag (IE bit) have been set to "1". After the HALT mode is released, if the master interrupt enable flag (MIE in PSW) has been set to "1", processing of the requested maskable interrupt is performed. If the master interrupt enable flag (MIE in PSW) has been reset to "0", the next instruction (following the instruction that set the HALT mode (that set bit 1 (HLT) of SBYCON to "1") is executed.

If the HALT mode is released by reset due to the $\overline{RES}$ pin input or overflow of the watchdog timer, the CPU will perform the reset processing.

(2) HOLD mode

When a high level is input to the HOLD pin after bit 5 (HOLD) of the peripheral control register (PRPHCON) is set to "1", the mode will change to the HOLD mode after the completion of the current instruction execution. Figure 3-3 shows the HOLD mode timing diagram.

In the HOLD mode, the clock (CPUCLK) supply to the CPU is terminated, but the clock (CPUCLK) is supplied to internal peripheral modules (TBC, general-purpose 8/16-bit timers, serial ports, etc.) so their operation continues. However, operation of the watchdog timer (WDT) is terminated. Because the CPU is halted, instructions are not executed. Instruction execution stops at the beginning of the next instruction (following the instruction that changed the mode to the HOLD mode).

If bus port functions (P0 to P4 set as secondary function outputs) are being used, the bus will be released during the HOLD mode.

If an interrupt occurs during the HOLD mode, because instructions are not being executed, interrupt processing will be suspended until the HOLD mode is released.

The HOLD mode is released when either a low level is input to the HOLD pin or the $\overline{RES}$ pin input causes a reset.

If a low level is input to the HOLD pin, instruction execution will resume starting from the next instruction (following the instruction that changed the mode to the HOLD mode). When an interrupt request occurs during the HOLD mode, if the interrupt is non-maskable, the non-maskable interrupt will be processed immediately after the HOLD mode is released. In the case of a maskable interrupt, if the corresponding interrupt enable flag (IE bit) and the master interrupt enable flag (MIE in PSW) have been set to "1", the maskable interrupt will be processed immediately after the HOLD mode is released. If multiple interrupt requests are generated, they are processed in order of priority.

If the HOLD mode is released by reset due to the $\overline{RES}$ pin input, the CPU will perform the reset processing.

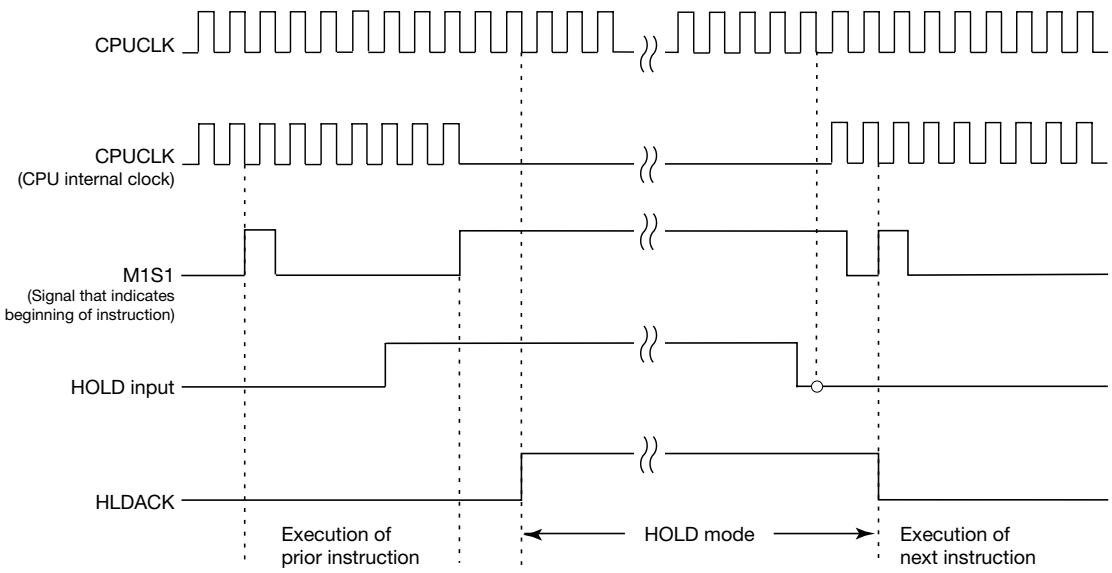


Figure 3-3 HOLD Mode Timing Diagram

(3) STOP mode

Setting the stop code acceptor (STPACP) to "1" by consecutively writing n5H, nAH (where n = 0 to F) and then setting bit 0 (STP) of the standby control register (SBYCON) to "1" will change the mode to the STOP mode.

In the STOP mode, the CPU and internal peripheral modules (TBC, WDT, general-purpose 8/16-bit timers, serial ports, etc.) are halted. However, when dual clocks are being used, the real-time counter (RTC) will operate as usual.

Because the clock supply to the CPU is halted, instructions are not executed. Instruction execution stops at the beginning of the next instruction (following the instruction that set bit 0 (STP) of SBYCON to "1").

The STOP mode is released when either an interrupt occurs or input to the $\overline{\text{RES}}$ pin causes a reset.

When the STOP mode is released due to an interrupt request, if the interrupt is non-maskable, the STOP mode is released unconditionally, and the CPU processes the non-maskable interrupt.

In the case of a maskable interrupt, the interrupt is released if the interrupt request flag (IRQ bit) and the interrupt enable flag (IE bit) have been set to "1".

During the STOP mode, the following factors generate maskable interrupt requests.

- Interrupt caused by input of the valid edge to an external interrupt pin (EXINT0 to EXINT5)
- Interrupt caused by real-time counter output (when dual clocks are used)

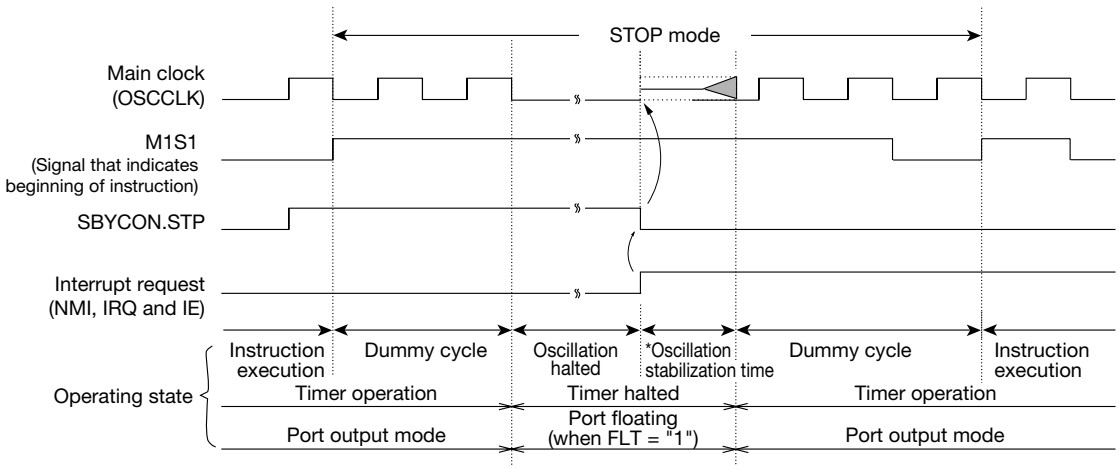
After the STOP mode is released, if the master interrupt enable flag (MIE in PSW) has been set to "1", processing of the requested maskable interrupt is performed.

If the master interrupt enable flag (MIE in PSW) has been reset to "0", the next instruction (following the instruction that set the STOP mode (that set bit 0 (STP) of SBYCON to "1") is executed. However, if the STOP mode has been set during the processing of a non-maskable interrupt routine, the STOP mode can be released by an interrupt request. After being released, the next instruction in the non-maskable interrupt routine (following the instruction that changed the mode to the STOP mode) will be executed. If interrupt priority is set (bit 7 (MIPF) of EXI2CON set to "1") and the STOP mode is set during a high priority interrupt routine, a low priority interrupt request can release the STOP mode. However, after release the low priority interrupt is suspended and the next instruction in the high priority interrupt routine will be executed.

If an interrupt request from the high-speed STOP mode (main clock oscillation terminated) causes the STOP mode to be released, operation will continue after waiting for the oscillation stabilization time of the main clock (OSCCLK) as set by SBYCON. The STOP mode can also be entered while the main clock continues to oscillate (quick activating STOP mode). In this case, when returning from the STOP mode, activation is possible without waiting for the oscillation stabilization time of the main clock.

Figure 3-4 shows the STOP mode timing diagram.

If the STOP mode is released by reset due to the $\overline{\text{RES}}$ pin input, the CPU will perform the reset processing. If the $\overline{\text{RES}}$ pin input is to be used to release the STOP mode with main clock oscillation halted, apply a low level to the $\overline{\text{RES}}$ pin until the main clock oscillation stabilizes.



* Oscillation stabilization time is the time until the main clock starts oscillating, plus the time of the number of clocks set by OST0 and OST1.

**Figure 3-4 STOP Mode Timing Diagram
(When released by an interrupt)**

3.3 Reset Function

The MSM66577 family is reset by the following four factors.

- Low-level input to the $\overline{RES}$ input pin
- Execution of a break (BRK) instruction
- Overflow of the watchdog timer (WDT)
- Opcode trap (OPTRP) due to execution of invalid instruction

Resets caused to the above four factors are processed in the same way except that the address of the vector address to be loaded in the program counter is different.

Table 3-3 lists the vector addresses for each reset factor.

Table 3-3 Vector Address for Each Reset Factor

Reset factor	Vector address [H]
Reset caused by low level input to the $\overline{RES}$ input pin	0000
Reset caused by execution of BRK instruction	0002
Reset caused by overflow of watchdog timer	0004
Reset caused by opcode trap	0006

During the reset processing, arithmetic registers, control registers, mode registers, etc. are initialized, and the contents of the address pointed to by the vector address is loaded into the program counter.

For the initial values of different registers, refer to Chapter 21, "Special Function Registers (SFRs)".

Reset has priority over all other processing (interrupt processing and instruction execution). Since all processing is aborted, register and RAM contents at that time cannot be guaranteed.

[Note]
If the $\overline{RES}$ pin input is to be used to for reset, apply a low level at the $\overline{RES}$ pin until the main clock oscillation stabilizes.

The Flash ROM version is reset by the supply voltage sense reset function when the power supply voltage is dropped, in the same way that the MSM66577 family is reset by low level input to the $\overline{RES}$ input pin. The supply voltage sense reset function is implemented when the supply voltage for the version operating in the range of 4.5 to 5.5 V is 3.0 V or less and the supply voltage for the version operating in the range of 3.0 to 3.6 V is 1.5 V or less. The reset function is not implemented during the STOP mode (only when oscillation clock is terminated).

Figure 3-5 shows an example of reset pin connection. Table 3-4 lists that status of I/O ports during reset.

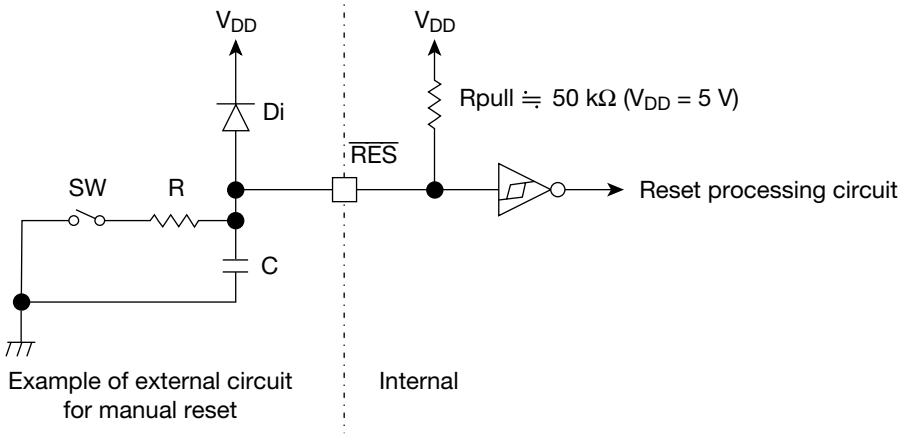


Figure 3-5 Reset Pin Connection Example

Table 3-4 I/O Port Status During Reset

Name	Low level $\overline{EA}$ pin P3_0, P3_1	High level $\overline{EA}$ pin P3_0, P3_1	Other ports
Status	Pulled-up	High impedance	High impedance

[Note]

If the $\overline{EA}$ pin is at a low level, after reset P0, P1, P2, P3_0, P3_1 and P4 automatically change to secondary function output states (bus port function).

Chapter 4

Memory Control Functions

4. Memory Control Functions

4.1 Overview

There are two independent memory spaces, the program memory space and the data memory space. The following three functions make the memory functions easier to use.

- ROM Window Function : This function enables various instructions that have been stored in the data memory space to also be used by the program in the program memory space.
- READY Function : If both memory spaces are to be used as external memory, this function allows the program to insert wait cycles into the external memory timing, according to the access times of the external memory.
- WAIT Function : This function enables an external device to control the insertion of wait cycles.

4.2 Memory Control Function Registers

Table 4-1 lists a summary of the SFRs for memory control functions.

Table 4-1 Summary of SFRs for Memory Control Functions

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
000B	ROM Window Register	ROMWIN	—	R/W	8	Undefined	4-2
000C☆	ROM Ready Control Register	ROMRDY	—	R/W	8	8B	4-4
000D☆	RAM Ready Control Register	RAMRDY	—	R/W	8	FF	4-5
0015☆	Peripheral Control Register	PRPHCON	—	R/W	8	8C	15-2

[Notes]

1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

4.3 ROM Window Function

The ROM window function reads the contents of the program memory space specified by the ROM window register (ROMWIN), located in the SFR area, by using the same address in the data memory space as a window.

In other words, when the ROM window function is enabled and an instruction that accesses (reads) the data memory space is executed, instead of accessing (reading) data in the data memory space, data will be accessed (read) at the same addresses in the segment that is specified by TSR in the program memory space.

Compared to the number of instruction cycles to be required to access normal data memory, accessing the ROM window once requires additional 3 cycles for a byte instruction and additional 6 cycles for a word instruction.

[Note]
If the ROM window function is enabled and a write instruction is executed, that result will not be guaranteed. However, in this case additional cycles will not be added.

- ROM Window Register (ROMWIN)
  The ROM window register (ROMWIN) is an 8-bit register. The lower 4 bits indicate the start address of the ROM window and the upper 4 bits indicate the end address of the ROM window. (Bits 4 and 5 of the upper 4 bits must be written as "1"s.) When 64KB of the program memory space is represented in hexadecimal number (HEX), each of above 4-bit registers specifies the upper 1 digit of 4 digits. If the value of the lower 4 bits is all zeros, the ROM window function will not operate.

  Figure 4-1 shows the configuration of ROMWIN.

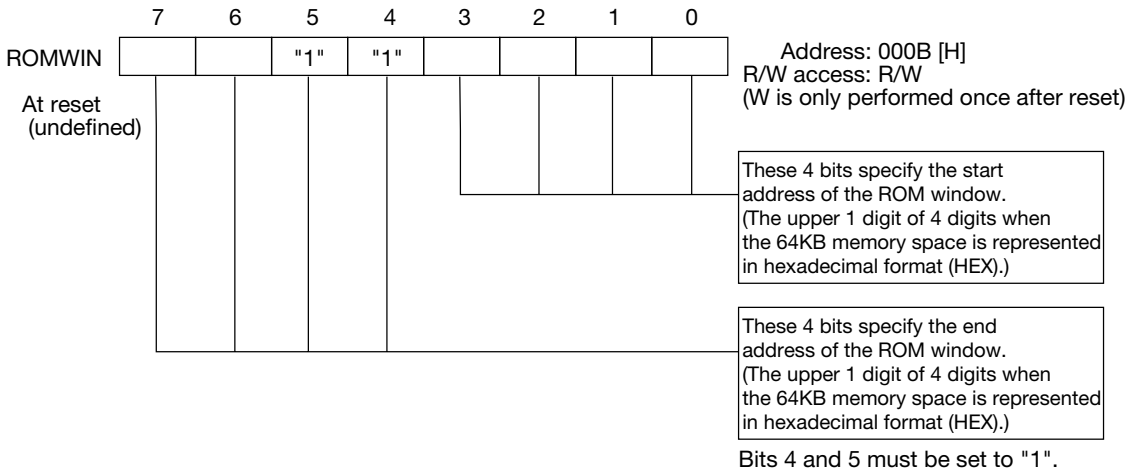


Figure 4-1 ROMWIN Configuration

If internal RAM is located in the data memory area specified as the ROM window, the data memory's internal RAM will have priority.

The data memory space specified as the ROM window area cannot be used as normal external data memory.

The ROM window start address is 1200H or above for segment 0, and 1000H or above for segments 1 to 15. The end address can be selected among the four end addresses listed in Table 4-2.

Table 4-2 End Address List

ROMWIN		End address [H]
Bit 7	Bit 6	
0	0	3FFF
0	1	7FFF
1	0	BFFF
1	1	FFFF

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ROMWIN is undefined and the ROM window function does not operate.

ROMWIN can be written to once after reset. Additional writing attempts will be ignored. Therefore, after the ROM window function has been set it can only be modified after a reset. ROMWIN can be read as many times as desired.

[Note]
The relative sizes of the start address "X" and the end address "Y" written to ROMWIN are not evaluated by the hardware. Therefore, be sure that $X \leq Y$ within the program.

4.4 READY Function

So that memory and general-purpose ICs with slow access speeds can be connected externally, wait cycles can be specified to be inserted during external memory accesses. There are two registers that specify the number of wait cycles, the ROM ready control register (ROMRDY) and the RAM ready control register (RAMRDY).

ROMRDY specifies wait cycles when the external ROM mode is used for the program memory space. By setting the IRORDY flag to "1", the same wait cycles specified for the external ROM are also applied to the internal ROM.

RAMRDY specifies wait cycles when the data memory space is extended externally. Memory can be divided into the two areas of address 0000H to 7FFFH and 8000H to FFFFH, and wait cycles can be specified for each area.

Table 4-3 lists the number of wait cycles that can be specified for RAMRDY and ROMRDY.

Table 4-3 Wait Cycles

Control register	Number of wait cycles to be inserted
ROMRDY	0 to 3
RAMRDY	0 to 7

4.4.1 ROM Ready Control Register (ROMRDY)

The ROM ready control register (ROMRDY) consists of two bits. ROMRDY specifies the number of wait cycles with bits 0 and 1 (ORDY0 and ORDY1) and insertion or no insertion of READY to the internal ROM with bit 2 (IRORDY).

ROMRDY can be read from and written to by the program. However, write operations are invalid for bits 3 and 7. Also, if writing to bits 4 to 6, they must be written as "0". When read, bits 3 and 7 are always "1" and bits 4 to 6 are "0".

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ROMRDY becomes 8BH and the largest number of wait cycles are set. Therefore, three wait cycles will be added and inserted when external program memory is accessed.

Figure 4-2 shows the configuration of ROMRDY.

	7	6	5	4	3	2	1	0
ROMRDY	—	"0"	"0"	"0"	—	IRORDY	ORDY1	ORDY0
At reset	1	0	0	0	1	0	1	1

Address: 000C [H]
R/W access: R/W

ORDY		Number of wait cycles to be added and inserted during an external program memory access
1	0	
0	0	0 cycles
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

IRORDY	Internal ROM READY
0	Not inserted
1	Inserted

"—" indicates a nonexistent bit.
When read, its value will be "1".

"0" indicates that the value "0" must always be written to this bit. When read, its value will be "0".

1 cycle is 1 CPUCLK.

Figure 4-2 ROMRDY Configuration

4.4.2 RAM Ready Control Register (RAMRDY)

The RAM ready control register (RAMRDY) consists of 6 bits. Bits 0 to 2 (ARDY00 to ARDY02) of RAMRDY specify the number of wait cycles for the external RAM area from 0000H to 7FFFH. Bits 4 to 6 (ARDY10 to ARDY12) specify the number of wait cycles for the external RAM area from 8000H to FFFFH. The number of wait cycles is uniform for all segments and settings are divided into the two areas of 0000H to 7FFFH (segment 0 is 1200H to 7FFFH) and 8000H to FFFFH.

RAMRDY can be read from and written to by the program. However, write operations are invalid for bits 3 and 7. When read, bits 3 and 7 are always "1".

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), RAMRDY becomes FFH and the largest number of wait cycles are set. Therefore, seven wait cycles will be added and inserted when external data memory is accessed.

Figure 4-3 shows the configuration of RAMRDY.

[Note]
In contrast to an internal data memory access, when external data memory is accessed, 2 or 3 cycles are automatically inserted for each 1 byte access. RAMRDY specifies the number of cycles to be inserted in addition to the 2 or 3 cycles that are inserted automatically inserted.

	7	6	5	4	3	2	1	0
RAMRDY	—	ARDY12	ARDY11	ARDY10	—	ARDY02	ARDY01	ARDY00
At reset	1	1	1	1	1	1	1	1

Address: 000D [H]
R/W access: R/W

ARDY0			Number of wait cycles to be added and inserted when accessing external data memory 0000H to 7FFFH
2	1	0	
0	0	0	0 cycles
0	0	1	1 cycle
0	1	0	2 cycles
0	1	1	3 cycles
1	0	0	4 cycles
1	0	1	5 cycles
1	1	0	6 cycles
1	1	1	7 cycles

ARDY1			Number of wait cycles to be added and inserted when accessing external data memory 8000H to FFFFH
2	1	0	
0	0	0	0 cycles
0	0	1	1 cycle
0	1	0	2 cycles
0	1	1	3 cycles
1	0	0	4 cycles
1	0	1	5 cycles
1	1	0	6 cycles
1	1	1	7 cycles

"—" indicates a nonexistent bit.
When read, its value will be "1".

1 cycle is 1 CPUCLK.

Figure 4-3 RAMRDY Configuration

4.5 WAIT Function

When accessing the external data memory area, in addition to the READY function that inserts wait cycles from the CPU, there is a WAIT function that can insert wait cycles via control from the external device. (This is applicable only to the data memory space.)

At the falling edge of CPUCLK shown in Figure 4-4 and 4-5, the high level of pin P11_0/WAIT is sampled and wait cycles are inserted into $\overline{WR}$ and $\overline{RD}$ strobe signals.

As shown in the sample timing of Figure 4-4, the WAIT function is released 1 tϕw after pin P11_0/WAIT is sample twice consecutively at a low level.

If the WAIT function is used with the READY function, the wait time that is largest will be valid.

In order to use the WAIT function, set bit 6 (WAIT) of the peripheral control register (PRPHCON) to "1". (Refer to Chapter 15, "Peripheral Functions".)

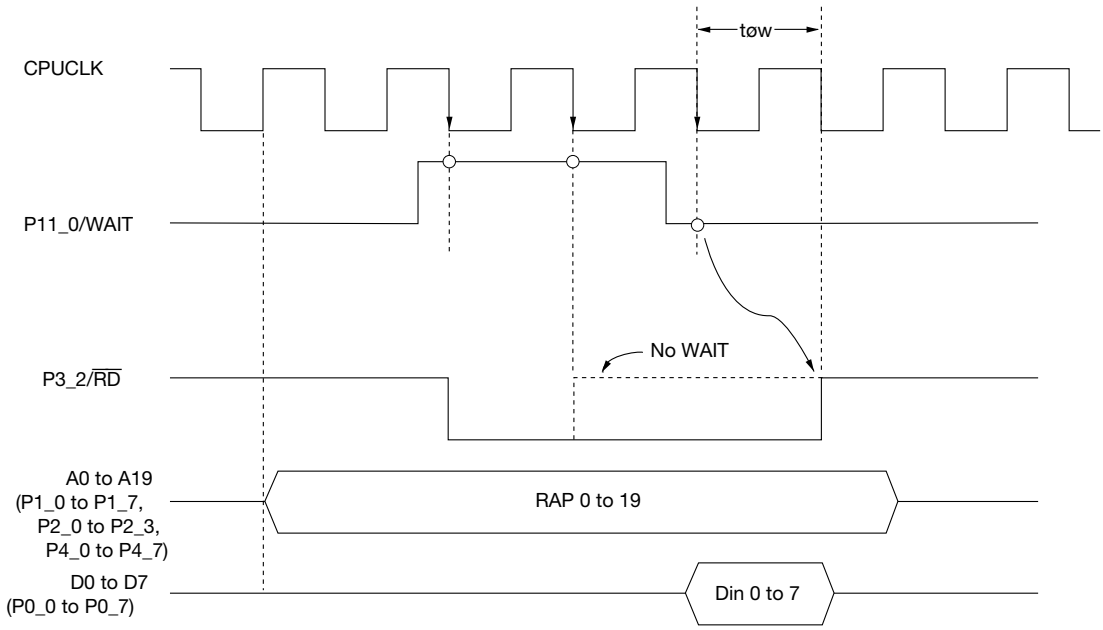


Figure 4-4 Sample Timing When Using the WAIT Function (Separate Bus Type)

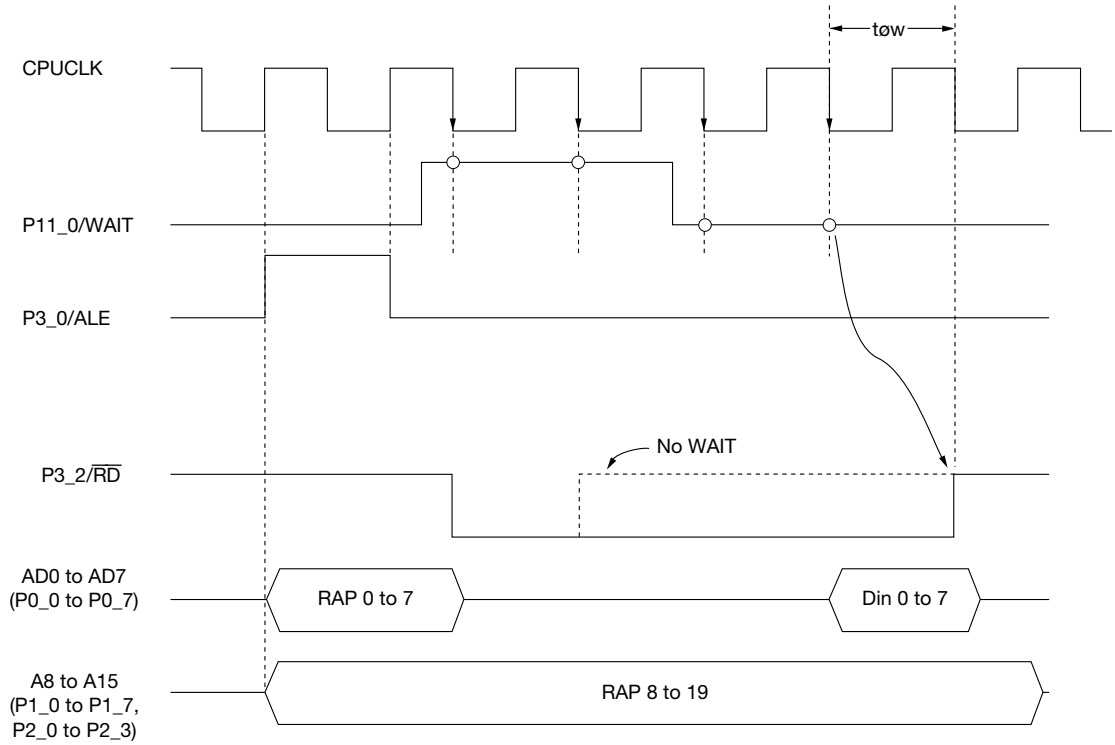


Figure 4-5 Sample Timing When Using the WAIT Function (Multiplexed Bus Type)

Chapter 5

Port Functions

5. Port Functions

5.1 Overview

The MSM66577 family has 14 sets of I/O port from P0 to P11, P14 and P15 (74 ports) and 1 set of input-only port at P12 (8 ports).

Each individual bit of all the I/O ports can be specified as input or output. All I/O ports have internal pull-up resistors that can be programmed for each individual bit.

The 3 sets of P0, P3 and P11 (16 ports) are capable of providing the sink current of 10 mA for driving LEDs.

If configured as inputs, the pins are high impedance inputs. If configured as outputs, they are push-pull outputs. In addition to the port function, some ports are assigned an internal function (secondary function).

Table 5-1 shows Port Function Summary.

Table 5-1 Port Function Summary

Port name	Pin	Type	Number	I/O	Secondary function
Port 0*	P0_0 to P0_7	A	8	I/O	External memory access address data bus AD0 to AD7 (I/O)
Port 1	P1_0 to P1_7	B	8	I/O	External memory access A8 to A15 (output)
Port 2	P2_0 to P2_3	B	4	I/O	External memory access A16 to A19 (output)
Port 3*	P3_0	B	1	I/O	External memory access ALE (output)
	P3_1	B	1	I/O	External program memory access $\overline{\text{PSEN}}$ (output)
	P3_2	C	1	I/O	External data memory access $\overline{\text{RD}}$ (output)
	P3_3	C	1	I/O	External data memory access $\overline{\text{WR}}$ (output)
Port 4	P4_0 to P4_7	B	8	I/O	External memory access A0 to A7 (output)
Port 5	P5_4, P5_5	D	2	I/O	Capture/compare CPCM0, CPCM1 (I/O)
	P5_6	D	1	I/O	Timer 0 timer output TM0OUT (output)
	P5_7	D	1	I/O	Timer 0 external event input TM0EVT (input)
Port 6	P6_0 to P6_3	D	4	I/O	External interrupt EXINT0 to EXINT3 (input)
	P6_4	D	1	I/O	Timer 1 external event input TM1EVT (input)
	P6_5	D	1	I/O	Timer 1 timer output TM1OUT (output)
	P6_6	D	1	I/O	Timer 2 external event input TM2EVT (input)
	P6_7	D	1	I/O	Timer 2 timer output TM2OUT (output)
Port 7	P7_6, P7_7	D	2	I/O	PWM output PWM0OUT, PWM1OUT (output)
Port 8	P8_0	D	1	I/O	SIO1 receive data input RXD1 (input)
	P8_1	D	1	I/O	SIO1 transmit data output TXD1 (output)
	P8_2	D	1	I/O	SIO1 receive clock RXC1 (I/O)
	P8_3	D	1	I/O	SIO1 transmit clock TXC1 (I/O)
	P8_4	D	1	I/O	Timer 4 timer output TM4OUT (output)
	P8_6, P8_7	D	2	I/O	PWM output PWM2OUT, PWM3OUT (output)
Port 9	P9_0 to P9_3	D	4	I/O	External interrupt EXINT4 to EXINT7 (input)
	P9_7	D	1	I/O	HOLD mode transfer output HLDACK (output)
Port 10	P10_3	D	1	I/O	SIO4 transmit-receive clock SIOCK4 (I/O)
	P10_4	D	1	I/O	SIO4 receive data output SIOO4 (output)
	P10_5	D	1	I/O	SIO4 transmit data input SIOI4 (input)
Port 11*	P11_0	D	1	I/O	External data memory access WAIT (input)
	P11_1	D	1	I/O	HOLD mode request input HOLD (input)
	P11_2	D	1	I/O	Main clock pulse output CLKOUT (output)
	P11_3	D	1	I/O	Subclock pulse output XTOUT (output)
Port 12	P12_0 to P12_7	F	8	I	A/D converter analog input AI0 to AI7 (input)
Port 14	P14_0	D	1	I/O	SIO5 transmit-receive clock SIOCK5 (I/O)
	P14_1	D	1	I/O	SIO5 transmit data output SIOO5 (output)
	P14_2	D	1	I/O	SIO5 receive data input SIOI5 (input)
	P14_6, P14_7	E	2	I/O	D/A converter analog output AO0, AO1 (output)
Port 15	P15_0	D	1	I/O	SIO6 transmit-receive data input RXD6 (input)
	P15_1	D	1	I/O	SIO6 transmit data output TXD6 (output)
	P15_2	D	1	I/O	SIO6 receive clock RXC6 (I/O)
	P15_3	D	1	I/O	SIO6 transmit clock TXC6 (I/O)

*Ports marked with an asterisk are capable of providing the sink current of 10 mA.

5.2 Hardware Configuration of Each Port

Ports (P0 to P12, P14, and P15) are divided into six categories, corresponding to each function.

5.2.1 Type A (P0)

The type A port has a secondary function and functions as an I/O pin. Depending on the state of the port mode registers (P0IOn) and the port secondary function control registers (P0SFn), the port configuration is switched between input, pulled-up input, output, and secondary function I/O (external memory data I/O in a separate bus type, and external memory data I/O and address output in a multiplexed bus type).

Because type A ports access external program memory as a secondary function, the port status is determined by the status of the $\overline{EA}$ pin (that specifies external memory access).

When reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow or an opcode trap), the pin status will be as follows:

$\overline{EA}$ pin status	Port initial status
H	High impedance input port
L	Secondary function I/O port

Figure 5-1 shows the type A configuration.

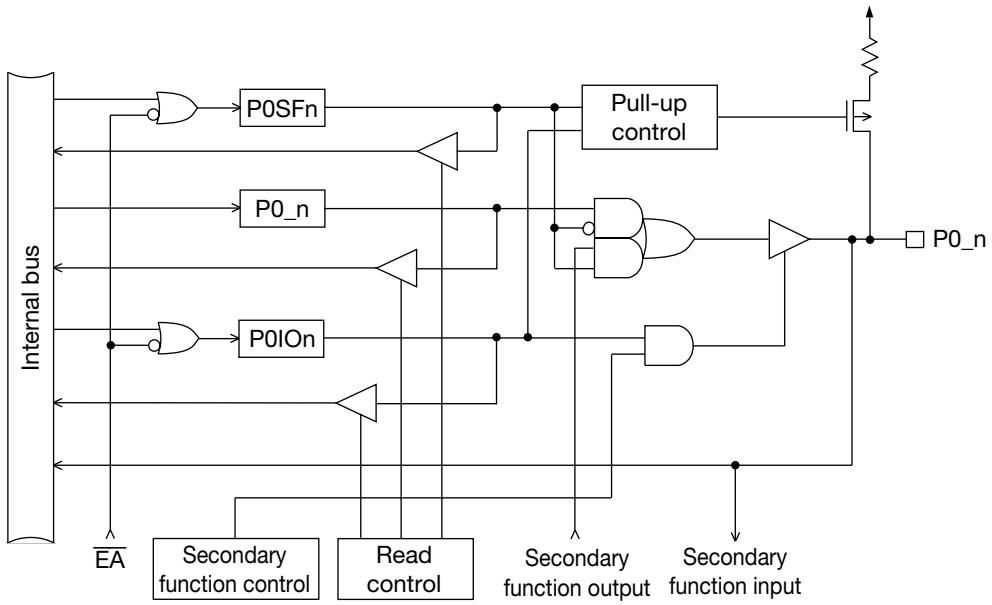


Figure 5-1 Type A Configuration

5.2.2 Type B (P1, P2, P3_0, P3_1, P4)

The type B port has a secondary function and functions as an I/O pin. Depending on the state of the port mode registers (PmIOn) and the port secondary function control registers (PmSFn), the port configuration is switched between input, pulled-up input, output, and secondary function output (external memory access).

Because type B ports access external program memory as a secondary function, the port status is determined by the status of the $\overline{EA}$ pin (that specifies external memory access).

When reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow or an opcode trap), the pin status will be as follows:

$\overline{EA}$ pin status	Port initial status
H	High impedance input port
L	Secondary function I/O port

Figure 5-2 shows the type B configuration.

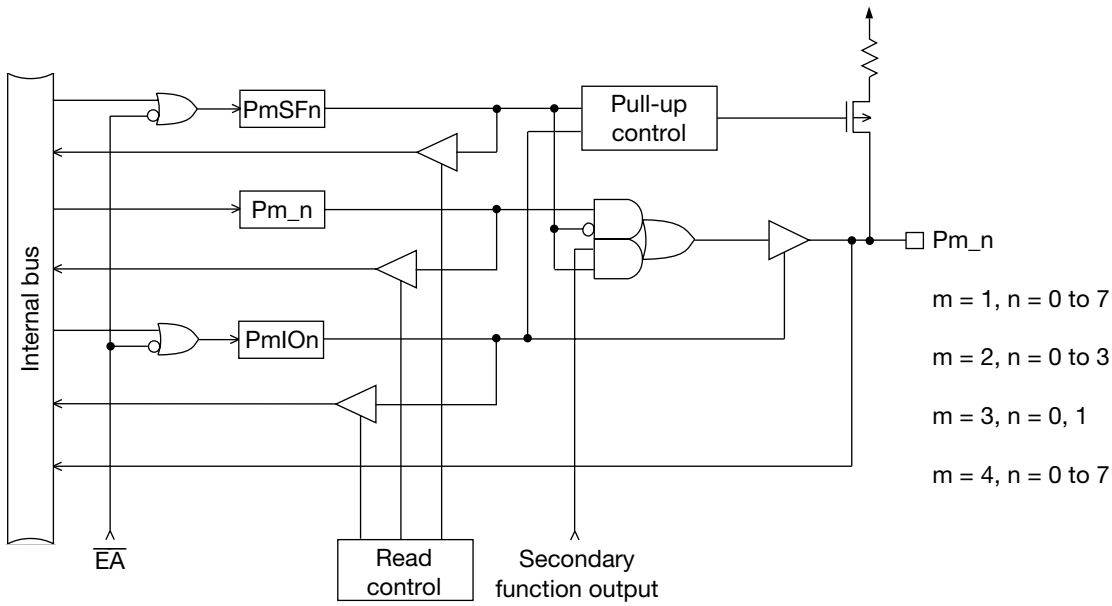


Figure 5-2 Type B Configuration

5.2.3 Type C (P3_2, P3_3)

The type C port has a secondary function. Depending on the state of the port mode registers (P3IOn) and the port secondary function control registers (P3SFn), the port configuration is switched between input, pulled-up input, output, and secondary function output (external memory access).

When reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of P3IOn and P3SFn is "0" and the port will be configured as a high impedance input port.

Figure 5-3 shows the type C configuration.

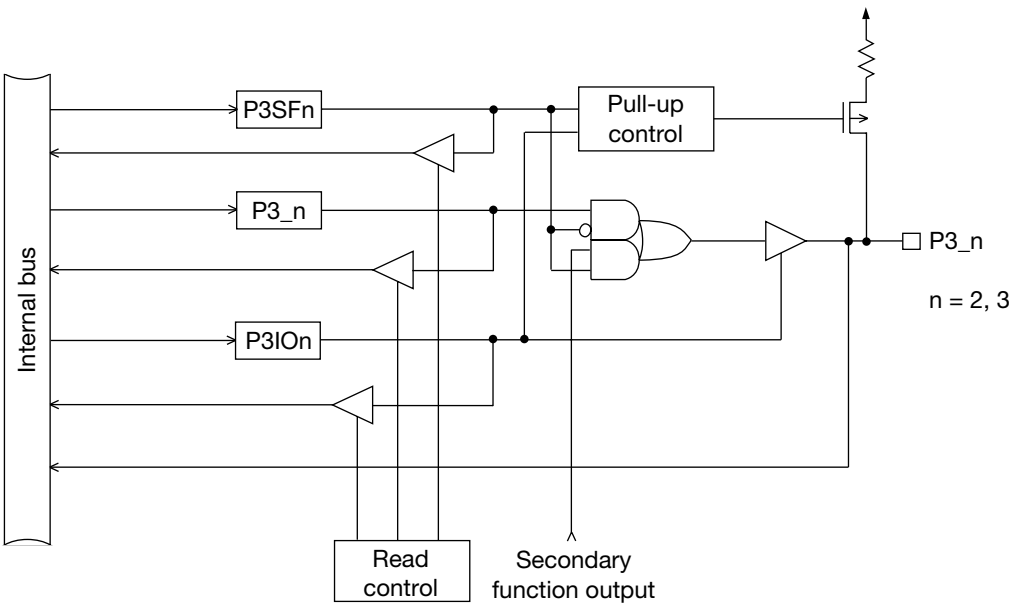


Figure 5-3 Type C Configuration

5.2.4 Type D (P5, P6, P7, P8, P9, P10, P11, P14_0 to P14_2, P15)

The type D port has a secondary function. Depending on the state of the port mode registers (PmIOn) and the port secondary function control registers (PmSFn), the port configuration is switched between input (primary/secondary function), pulled-up input (primary/secondary function), output, and secondary function output.

When reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of PmIOn and PmSFn is "0" and the port will be configured as a high impedance input port.

Figure 5-4 shows the type D configuration.

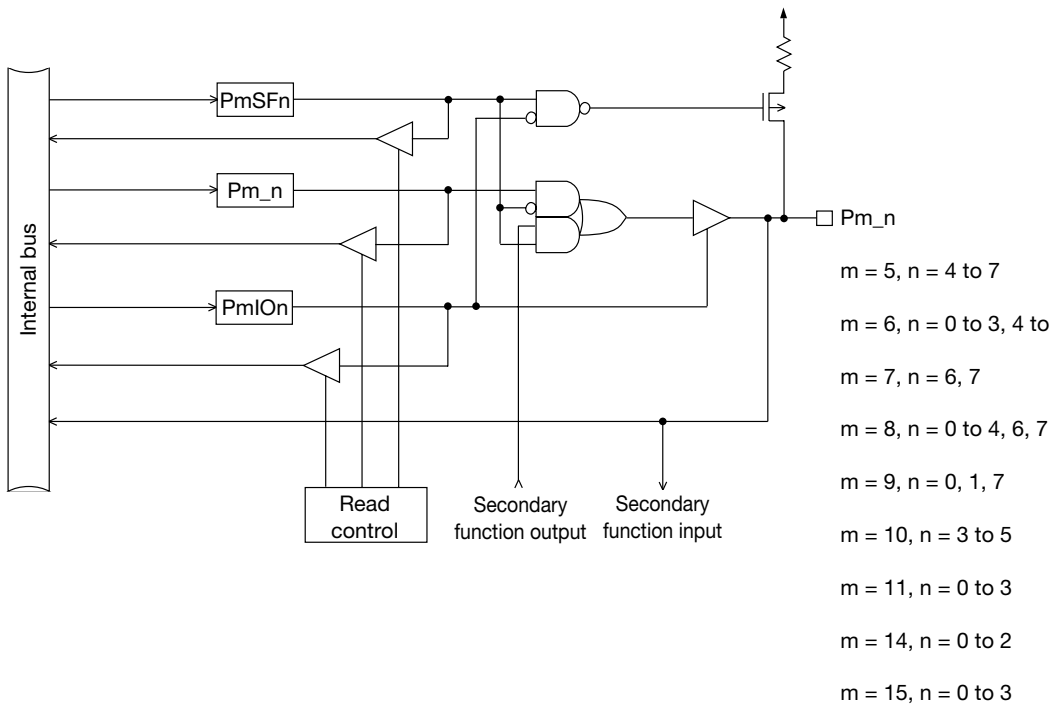


Figure 5-4 Type D Configuration

5.2.5 Type E (P14_6, P14_7)

The type E port has a secondary function. Depending on the state of the port mode registers (PmIOn) and the port secondary function control registers (PmSFn), the port configuration is switched between input (primary/secondary function), pulled-up input (primary/secondary function), output, and secondary function output.

When reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of PI4IOn and PI4SFn is "0" and the port will be configured as a high impedance input port.

Figure 5-5 shows the type E configuration.

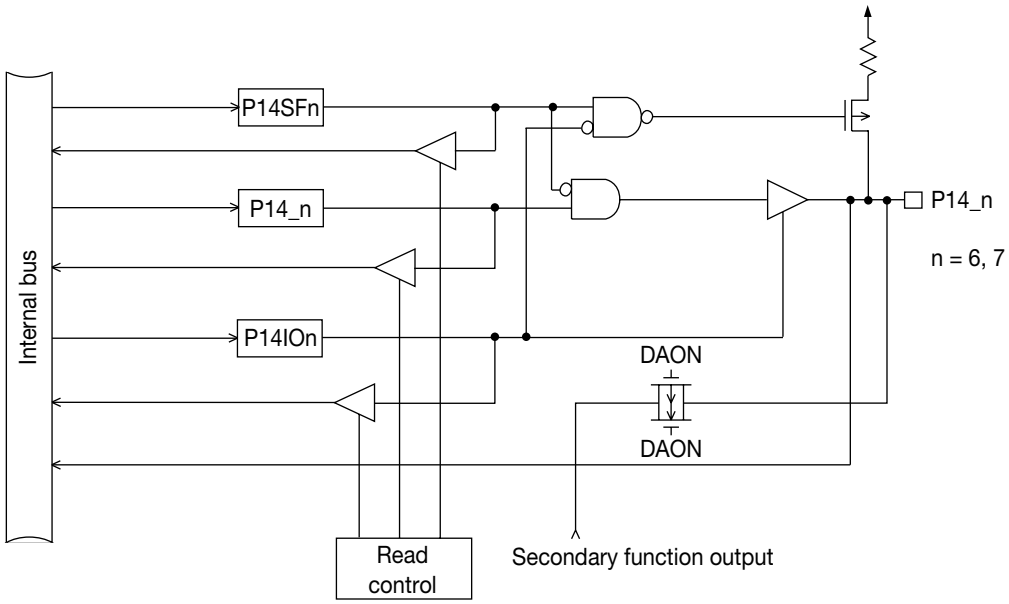


Figure 5-5 Type E Configuration

5.2.6 Type F (P12)

The type F port has a secondary function input, but is an input-only port that is not assigned a port mode register (PnIO) and a port secondary function control register (PnSF).
P12 also functions as the analog input of the A/D converter.
Figure 5-6 shows the type F configuration.

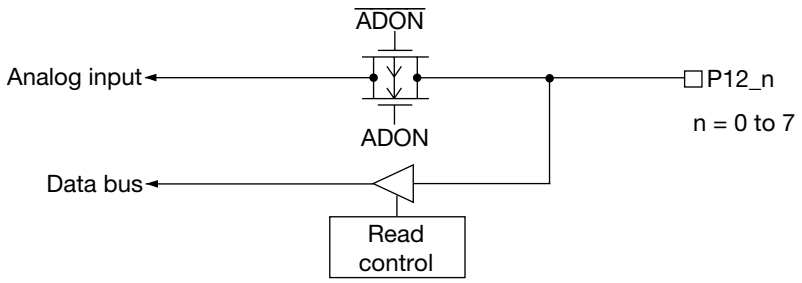


Figure 5-6 Type F Configuration

5.3 Port Registers

There are three types of port control registers:

- Port data registers (Pn: n = 0 to 12, 14, 15)
- Port mode registers (PnIO: n = 0 to 11, 14, 15)
- Port secondary function control registers (PnSF: n = 0 to 11, 14, 15)

These registers are allocated as SFRs.
Table 5-2 lists a summary of the port control SFRs.

Table 5-2 Port Control SFR Summary (1/2)

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 operation	Initial value [H]	Reference page
0018	Port 0 data register	P0	—	R/W	8	00	5-12
0019	Port 1 data register	P1	—	R/W	8	00	5-14
001A☆	Port 2 data register	P2	—	R/W	8	00	5-16
001B☆	Port 3 data register	P3	—	R/W	8	00	5-18
001C	Port 4 data register	P4	—	R/W	8	00	5-20
001D☆	Port 5 data register	P5	—	R/W	8	00	5-22
001E	Port 6 data register	P6	—	R/W	8	00	5-24
001F☆	Port 7 data register	P7	—	R/W	8	00	5-26
00B8☆	Port 8 data register	P8	—	R/W	8	00	5-28
00B9☆	Port 9 data register	P9	—	R/W	8	00	5-30
00BA☆	Port 10 data register	P10	—	R/W	8	00	5-32
00BB☆	Port 11 data register	P11	—	R/W	8	00	5-34
00BC	Port 12 data register	P12	—	R	8	Undefined	5-36
00BE☆	Port 14 data register	P14	—	R/W	8	00	5-37
00BF☆	Port 15 data register	P15	—	R/W	8	00	5-39
0020	Port 0 mode register	P0IO	—	R/W	8	00/FF	5-12
0021	Port 1 mode register	P1IO	—	R/W	8	00/FF	5-14
0022☆	Port 2 mode register	P2IO	—	R/W	8	00/0F	5-16
0023☆	Port 3 mode register	P3IO	—	R/W	8	00/02	5-18
0024	Port 4 mode register	P4IO	—	R/W	8	00/FF	5-20
0025☆	Port 5 mode register	P5IO	—	R/W	8	00	5-22
0026	Port 6 mode register	P6IO	—	R/W	8	00	5-24
0027☆	Port 7 mode register	P7IO	—	R/W	8	00	5-26
00C0☆	Port 8 mode register	P8IO	—	R/W	8	00	5-28
00C1☆	Port 9 mode register	P9IO	—	R/W	8	00	5-30
00C2☆	Port 10 mode register	P10IO	—	R/W	8	00	5-32
00C3☆	Port 11 mode register	P11IO	—	R/W	8	00	5-34
00C4☆	Port 14 mode register	P14IO	—	R/W	8	00	5-37
00C5☆	Port 15 mode register	P15IO	—	R/W	8	00	5-39

Table 5-2 Port Control SFR Summary (2/2)

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 operation	Initial value [H]	Reference page
0028	Port 0 secondary function control register	P0SF	—	R/W	8	00/FF	5-12
0029	Port 1 secondary function control register	P1SF	—	R/W	8	00/FF	5-14
002A☆	Port 2 secondary function control register	P2SF	—	R/W	8	00/0F	5-16
002B☆	Port 3 secondary function control register	P3SF	—	R/W	8	00/03	5-18
002C	Port 4 secondary function control register	P4SF	—	R/W	8	00/FF	5-20
002D☆	Port 5 secondary function control register	P5SF	—	R/W	8	00	5-22
002E	Port 6 secondary function control register	P6SF	—	R/W	8	00	5-24
002F☆	Port 7 secondary function control register	P7SF	—	R/W	8	00	5-26
00C6☆	Port 14 secondary function control register	P14SF	—	R/W	8	00	5-37
00C7☆	Port 15 secondary function control register	P15SF	—	R/W	8	00	5-39
00C8☆	Port 8 secondary function control register	P8SF	—	R/W	8	00	5-28
00C9☆	Port 9 secondary function control register	P9SF	—	R/W	8	00	5-30
00CA☆	Port 10 secondary function control register	P10SF	—	R/W	8	00	5-32
00CB☆	Port 11 secondary function control register	P11SF	—	R/W	8	00	5-34

[Notes]

1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. Initial values may change depending upon the status of the $\overline{EA}$ pin (mode registers and secondary control registers for port 0 to port 4). Listings are in the order of $\overline{EA}$ = high-level/low-level.
4. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

5.3.1 Port Data Registers (Pn : n = 0 to 12, 14, 15)

Port data registers (Pn : n = 0 to 12, 14, 15) store the port output data.

Pn registers are allocated as SFRs and when reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), their value becomes 00H.

If an instruction to read Pn is executed, for ports specified as inputs, the pin status ("0" or "1") will be read. For ports specified as outputs, the Pn status ("0" or "1") will be read. If an instruction to write to Pn is executed, regardless whether the port is input or output, data will be written to Pn. Because P12 is an input-only port, only read instructions can be executed. If a read instruction is executed, the pin status ("0" or "1") will be read.

[Note]

If a bit specified as input by the port mode register (PnIO) is read, the pin status will be read. When writing data to a port data register (Pn), if read-modify-write instructions such as arithmetic, logical and bit manipulation instructions are used, the port data register (Pn) of the bit specified as an input will be overwritten.

5.3.2 Port Mode Registers (PnIO : n = 0 to 11, 14, 15)

Port mode registers (PnIO : n = 0 to 11, 14, 15) specify whether I/O ports are inputs or outputs.

PnIO registers are allocated as SFRs and when reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, opcode trap), their value becomes 00H and all ports will be set to the input mode. However, if the $\overline{EA}$ pin is at a low level, ports used to access external memory will automatically be set to the output mode.

Setting each individual bit of PnIO to "0" configures the input mode and "1" configures the output mode.

5.3.3 Port Secondary Function Control Registers (PnSF : n = 0 to 11, 14, 15)

Port secondary function control registers (PnSF : n = 0 to 11, 14, 15) specify the secondary function output for ports.

PnSF registers are allocated as SFRs and when reset (due to $\overline{RES}$ input, BRK instruction execution, watchdogtimer overflow, or opcode trap) their values become 00H and the primary function will be selected for all ports. However, if the $\overline{EA}$ pin is at a low level, ports used to access external memory will automatically be configured as secondary function outputs.

When the port is in input mode, if PnSF is set to "1", the input will be pulled-up. When the port is in output mode, if PnSF is set to "1", the secondary function output will be selected. The secondary function input does not depend upon PnSF, and can be read in the same manner as the primary function input with PnIO = 0.

Table 5-3 lists the port status due to the settings of the port mode register and the port secondary function control register.

Table 5-3 Port Settings

PnIO	PnSF	Function
0	0	Input (primary/secondary function)
0	1	Pulled-up input (primary/secondary function)
1	0	Output (primary function)
1	1	Output (secondary function)

If a port that is not assigned a secondary function is set to secondary function output (PnIO = 1, PnSF = 1), "0" will be output to that port.

Table 5-4 lists the values read when reading the port data register (Pn : n = 0 to 11, 14, 15) according to the settings of port mode register (PnIO) and port secondary control register (PnSF).

Table 5-4 Port Data Register Read Data

PnIO	PnSF	Read data
0	*	Pin status
1	0	Pn (value of port data register)
1	1	Output secondary function data

*: "0" or "1," n: 0 to 11, 14, 15

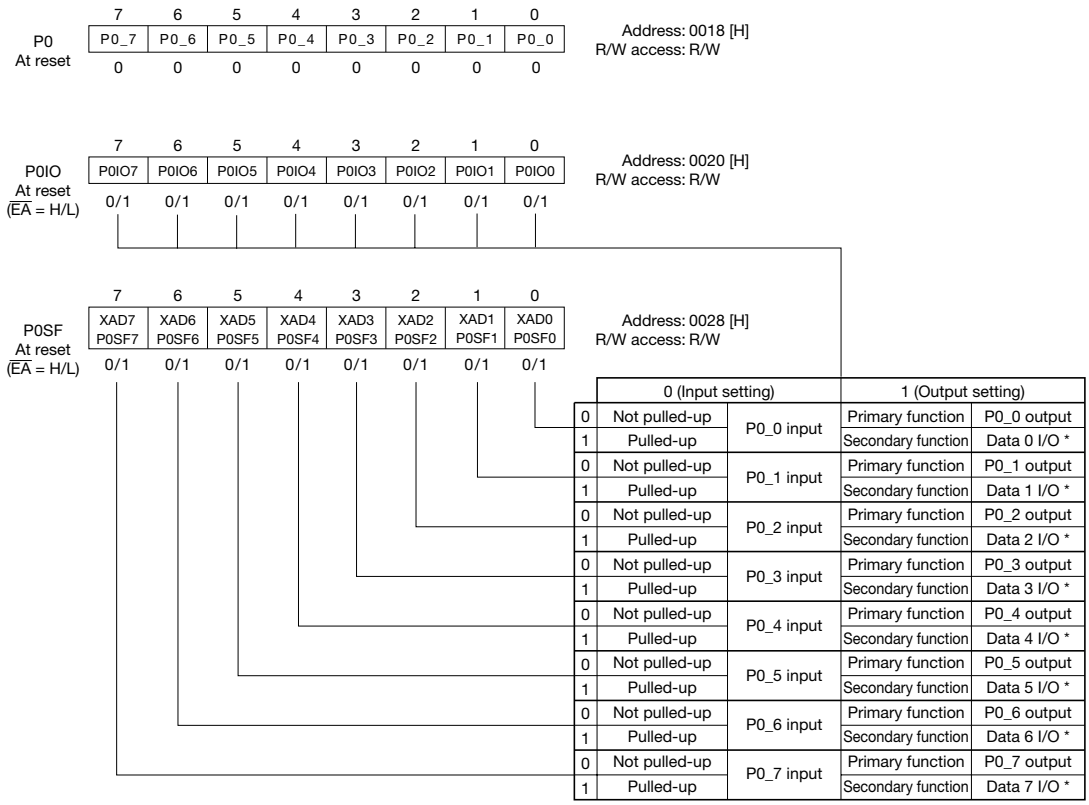
5.4 Port 0 (P0)

Port 0 is an 8-bit I/O port. Each individual bit can be specified as input or output by the port 0 mode register (P0IO). When output is specified (corresponding bits of P0IO = "1"), the value of the corresponding bits in the port 0 data register (P0) will be output from their appropriate pins.

In addition to its port function, P0 is assigned a secondary function (external memory data I/O when selecting a separate bus type, and external memory data I/O and address output when selecting a multiplexed bus type). If the secondary function is to be used, set the corresponding bits of the port 0 mode register (P0IO) and the port 0 secondary function control register (P0SF) to "1".

If the port is specified as an input (corresponding bits of P0IO = "0") and the port 0 secondary function control register (P0SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-7 shows the configuration of the port 0 data register (P0), port 0 mode register (P0IO) and the port 0 secondary function control register (P0SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P0_0 input	Primary function	P0_0 output
1	Pulled-up		Secondary function	Data 0 I/O *
0	Not pulled-up	P0_1 input	Primary function	P0_1 output
1	Pulled-up		Secondary function	Data 1 I/O *
0	Not pulled-up	P0_2 input	Primary function	P0_2 output
1	Pulled-up		Secondary function	Data 2 I/O *
0	Not pulled-up	P0_3 input	Primary function	P0_3 output
1	Pulled-up		Secondary function	Data 3 I/O *
0	Not pulled-up	P0_4 input	Primary function	P0_4 output
1	Pulled-up		Secondary function	Data 4 I/O *
0	Not pulled-up	P0_5 input	Primary function	P0_5 output
1	Pulled-up		Secondary function	Data 5 I/O *
0	Not pulled-up	P0_6 input	Primary function	P0_6 output
1	Pulled-up		Secondary function	Data 6 I/O *
0	Not pulled-up	P0_7 input	Primary function	P0_7 output
1	Pulled-up		Secondary function	Data 7 I/O *

"*" indicates data I/O in a separate bus type. In a multiplexed bus type, it will be data I/O and address output.

Figure 5-7 P0, P0IO, P0SF Configuration

Table 5-5 lists the data that is read, depending on the settings of P0IO and P0SF, when executing an instruction to read P0.

At reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the $\overline{EA}$ pin is at a high level, P0 will become a high impedance input port (P0IO = 00H, P0SF = 00H) and the contents of P0 will be 00H. If the $\overline{EA}$ pin is at a low level, P0 will be set as a secondary function I/O port (P0IO = FFH, P0SF = FFH) and the contents of P0 will be 00H.

Table 5-5 P0 Read Data

	P0IO	P0SF	Read data
P0_0	0	*	P0_0 pin state
	1	*	Value of bit 0 of P0 (port data register)
P0_1	0	*	P0_1 pin state
	1	*	Value of bit 1 of P0 (port data register)
P0_2	0	*	P0_2 pin state
	1	*	Value of bit 2 of P0 (port data register)
P0_3	0	*	P0_3 pin state
	1	*	Value of bit 3 of P0 (port data register)
P0_4	0	*	P0_4 pin state
	1	*	Value of bit 4 of P0 (port data register)
P0_5	0	*	P0_5 pin state
	1	*	Value of bit 5 of P0 (port data register)
P0_6	0	*	P0_6 pin state
	1	*	Value of bit 6 of P0 (port data register)
P0_7	0	*	P0_7 pin state
	1	*	Value of bit 7 of P0 (port data register)

"*" indicates "0" or "1"

[Note]

If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P0, depending on the settings of P0IO and P0SF, values will be read as listed in Table 5-5. The modified values will be written to P0 (port 0 data register).

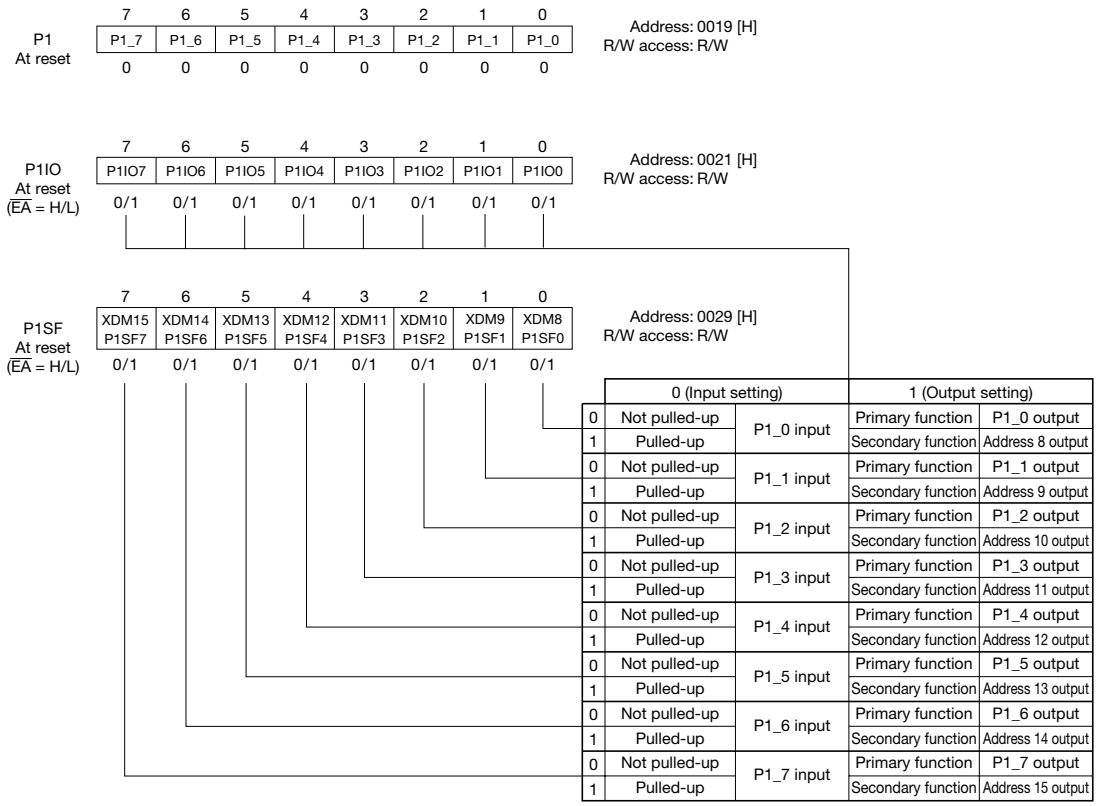
5.5 Port 1 (P1)

Port 1 is an 8-bit I/O port. Each individual bit can be specified as input or output by the port 1 mode register (P1IO). When output is specified (corresponding bits of P1IO = "1"), the value of the corresponding bits in the port 1 data register (P1) will be output from their appropriate pins.

In addition to its port function, P1 is assigned a secondary function (external memory address output). If the secondary function is to be used, set the corresponding bits of the port 1 mode register (P1IO) and the port 1 secondary function control register (P1SF) to "1".

If the port is specified as an input (corresponding bits of P1IO = "0") and the port 1 secondary function control register (P1SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-8 shows the configuration of the port 1 data register (P1), port 1 mode register (P1IO) and the port 1 secondary function control register (P1SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P1_0 input	Primary function	P1_0 output
1	Pulled-up		Secondary function	Address 8 output
0	Not pulled-up	P1_1 input	Primary function	P1_1 output
1	Pulled-up		Secondary function	Address 9 output
0	Not pulled-up	P1_2 input	Primary function	P1_2 output
1	Pulled-up		Secondary function	Address 10 output
0	Not pulled-up	P1_3 input	Primary function	P1_3 output
1	Pulled-up		Secondary function	Address 11 output
0	Not pulled-up	P1_4 input	Primary function	P1_4 output
1	Pulled-up		Secondary function	Address 12 output
0	Not pulled-up	P1_5 input	Primary function	P1_5 output
1	Pulled-up		Secondary function	Address 13 output
0	Not pulled-up	P1_6 input	Primary function	P1_6 output
1	Pulled-up		Secondary function	Address 14 output
0	Not pulled-up	P1_7 input	Primary function	P1_7 output
1	Pulled-up		Secondary function	Address 15 output

Figure 5-8 P1, P1IO, P1SF Configuration

Table 5-6 lists the data that is read, depending on the settings of P1IO and P1SF, when executing an instruction to read P1.

At reset (due to $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the $\overline{EA}$ pin is at a high level, P1 will become a high impedance input port (P1IO = 00H, P1SF = 00H) and the contents of P1 will be 00H. If the $\overline{EA}$ pin is at a low level, P1 will be set as a secondary function output port (P1IO = FFH, P1SF = FFH) and the contents of P1 will be 00H.

Table 5-6 P1 Read Data

	P1IO	P1SF	Read data
P1_0	0	*	P1_0 pin state
	1	*	Value of bit 0 of P1 (port data register)
P1_1	0	*	P1_1 pin state
	1	*	Value of bit 1 of P1 (port data register)
P1_2	0	*	P1_2 pin state
	1	*	Value of bit 2 of P1 (port data register)
P1_3	0	*	P1_3 pin state
	1	*	Value of bit 3 of P1 (port data register)
P1_4	0	*	P1_4 pin state
	1	*	Value of bit 4 of P1 (port data register)
P1_5	0	*	P1_5 pin state
	1	*	Value of bit 5 of P1 (port data register)
P1_6	0	*	P1_6 pin state
	1	*	Value of bit 6 of P1 (port data register)
P1_7	0	*	P1_7 pin state
	1	*	Value of bit 7 of P1 (port data register)

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P1, depending on the settings of P1IO and P1SF, values will be read as listed in Table 5-6. The modified values will be written to P1 (port 1 data register).

5.6 Port 2 (P2)

Port 2 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 2 mode register (P2IO). When output is specified (corresponding bits of P2IO = "1"), the value of the corresponding bits in the port 2 data register (P2) will be output from their appropriate pins.

In addition to its port function, P2 is assigned a secondary function (external memory address output). If the secondary function is to be used, set the corresponding bits of the port 2 mode register (P2IO) and the port 2 secondary function control register (P2SF) to "1".

If the port is specified as an input (corresponding bits of P2IO = "0") and the port 2 secondary function control register (P2SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-9 shows the configuration of the port 2 data register (P2), port 2 mode register (P2IO) and the port 2 secondary function control register (P2SF).

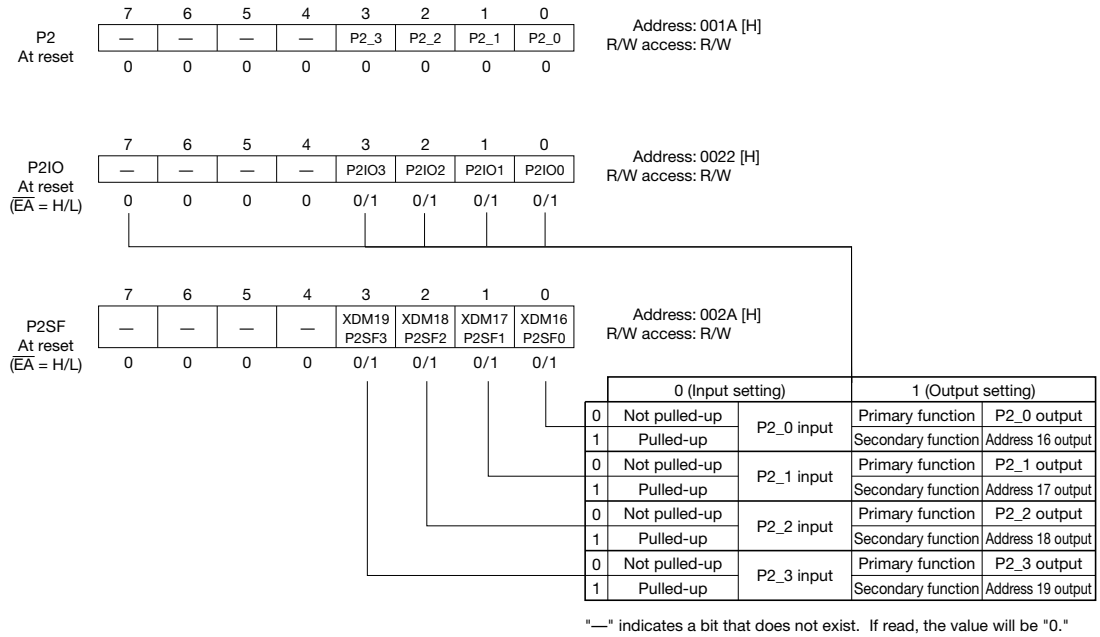


Figure 5-9 P2, P2IO, P2SF Configuration

Table 5-7 lists the data that is read, depending on the settings of P2IO and P2SF, when executing an instruction to read P2.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the $\overline{EA}$ pin is at a high level, P2 will become a high impedance input port (P2IO = 00H, P2SF = 00H) and the contents of P2 will be 00H. If the $\overline{EA}$ pin is at a low level, P2 will be set as a secondary function output port (P2IO = 0FH, P2SF = 0FH) and the contents of P2 will be 00H.

Table 5-7 P2 Read Data

	P2IO	P2SF	Read data
P2_0	0	*	P2_0 pin state
	1	*	Value of bit 0 of P2 (port data register)
P2_1	0	*	P2_1 pin state
	1	*	Value of bit 1 of P2 (port data register)
P2_2	0	*	P2_2 pin state
	1	*	Value of bit 2 of P2 (port data register)
P2_3	0	*	P2_3 pin state
	1	*	Value of bit 3 of P2 (port data register)

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P2, depending on the settings of P2IO and P2SF, values will be read as listed in Table 5-7. The modified values will be written to P2 (port 2 data register).

5.7 Port 3 (P3)

Port 3 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 3 mode register (P3IO). When output is specified (corresponding bits of P3IO = "1"), the value of the corresponding bits in the port 3 data register (P3) will be output from their appropriate pins.

In addition to its port function, P3 is assigned secondary functions (ALE, $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$ outputs). If a secondary function is to be used, set the corresponding bits of the port 3 mode register (P3IO) and the port 3 secondary function control register (P3SF) to "1".

If the port is specified as an input (corresponding bits of P3IO = "0") and the port 3 secondary function control register (P3SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-10 shows the configuration of the port 3 data register (P3), port 3 mode register (P3IO) and the port 3 secondary function control register (P3SF).

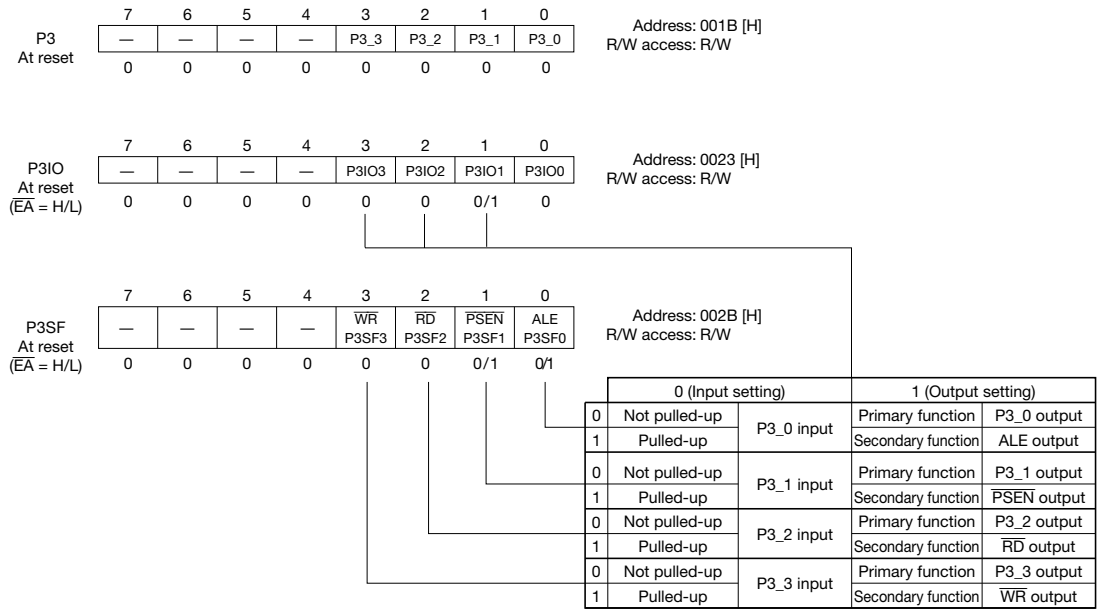


Figure 5-10 P3, P3IO, P3SF Configuration

Table 5-8 lists the data that is read, depending on the settings of P3IO and P3SF, when executing an instruction to read P3.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the $\overline{EA}$ pin is at a high level, P3 will become a high impedance input port (P3IO = 00H, P3SF = 00H) and the contents of P3 will be 00H. If the $\overline{EA}$ pin is at a low level, only P3_0 and P3_1 will be set as a secondary function I/O port (P3IO = 03H, P3SF = 03H) and the contents of P3 will be 00H.

Table 5-8 Read Data

	P3IO	P3SF	Read data
P3_0	0	*	P3_0 pin state
	1	*	Value of bit 0 of P3 (port data register)
P3_1	0	*	P3_1 pin state
	1	*	Value of bit 1 of P3 (port data register)
P3_2	0	*	P3_2 pin state
	1	*	Value of bit 2 of P3 (port data register)
P3_3	0	*	P3_3 pin state
	1	*	Value of bit 3 of P3 (port data register)

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P3, depending on the settings of P3IO and P3SF, values will be read as listed in Table 5-8. The modified values will be written to P3 (port 3 data register).

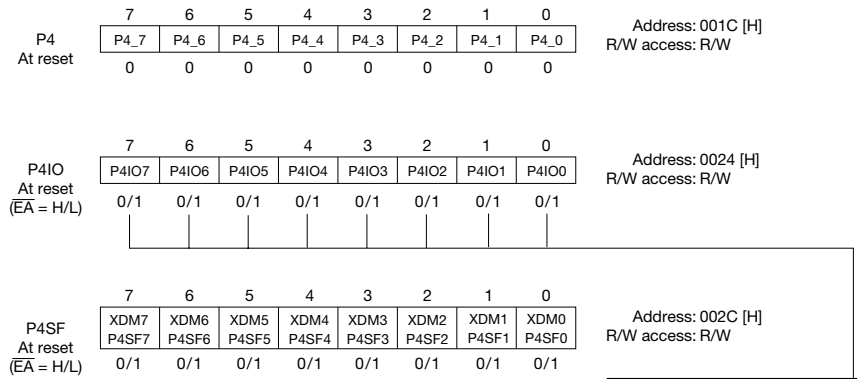
5.8 Port 4 (P4)

Port 4 is an 8-bit I/O port. Each individual bit can be specified as input or output by the port 4 mode register (P4IO). When output is specified (corresponding bits of P4IO = "1"), the value of the corresponding bits in the port 4 data register (P4) will be output from their appropriate pins.

In addition to its port function, P4 is assigned a secondary function (external memory address output when selecting a separate bus type). If the secondary function is to be used, set the corresponding bits of the port 4 mode register (P4IO) and the port 4 secondary function control register (P4SF) to "1".

If the port is specified as an input (corresponding bits of P4IO = "0") and the port 4 secondary function control register (P4SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-11 shows the configuration of the port 4 data register (P4), port 4 mode register (P4IO) and the port 4 secondary function control register (P4SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P4_0 input	Primary function	P4_0 output
1	Pulled-up		Secondary function	Address 0 output
0	Not pulled-up	P4_1 input	Primary function	P4_1 output
1	Pulled-up		Secondary function	Address 1 output
0	Not pulled-up	P4_2 input	Primary function	P4_2 output
1	Pulled-up		Secondary function	Address 2 output
0	Not pulled-up	P4_3 input	Primary function	P4_3 output
1	Pulled-up		Secondary function	Address 3 output
0	Not pulled-up	P4_4 input	Primary function	P4_4 output
1	Pulled-up		Secondary function	Address 4 output
0	Not pulled-up	P4_5 input	Primary function	P4_5 output
1	Pulled-up		Secondary function	Address 5 output
0	Not pulled-up	P4_6 input	Primary function	P4_6 output
1	Pulled-up		Secondary function	Address 6 output
0	Not pulled-up	P4_7 input	Primary function	P4_7 output
1	Pulled-up		Secondary function	Address 7 output

Figure 5-11 P4, P4IO, P4SF Configuration

Table 5-9 lists the data that is read, depending on the settings of P4IO and P4SF, when executing an instruction to read P4.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the $\overline{EA}$ pin is at a high level, P4 will become a high impedance input port (P4IO = 00H, P4SF = 00H) and the contents of P4 will be 00H. If the $\overline{EA}$ pin is at a low level, P4 will be set as a secondary function output port (P4IO = FFH, P4SF = FFH) and the contents of P4 will be 00H.

Table 5-9 P4 Read Data

	P4IO	P4SF	Read data
P4_0	0	*	P4_0 pin state
	1	*	Value of bit 0 of P4 (port data register)
P4_1	0	*	P4_1 pin state
	1	*	Value of bit 1 of P4 (port data register)
P4_2	0	*	P4_2 pin state
	1	*	Value of bit 2 of P4 (port data register)
P4_3	0	*	P4_3 pin state
	1	*	Value of bit 3 of P4 (port data register)
P4_4	0	*	P4_4 pin state
	1	*	Value of bit 4 of P4 (port data register)
P4_5	0	*	P4_5 pin state
	1	*	Value of bit 5 of P4 (port data register)
P4_6	0	*	P4_6 pin state
	1	*	Value of bit 6 of P4 (port data register)
P4_7	0	*	P4_7 pin state
	1	*	Value of bit 7 of P4 (port data register)

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P4, depending on the settings of P4IO and P4SF, values will be read as listed in Table 5-9. The modified values will be written to P4 (port 4 data register).

5.9 Port 5 (P5)

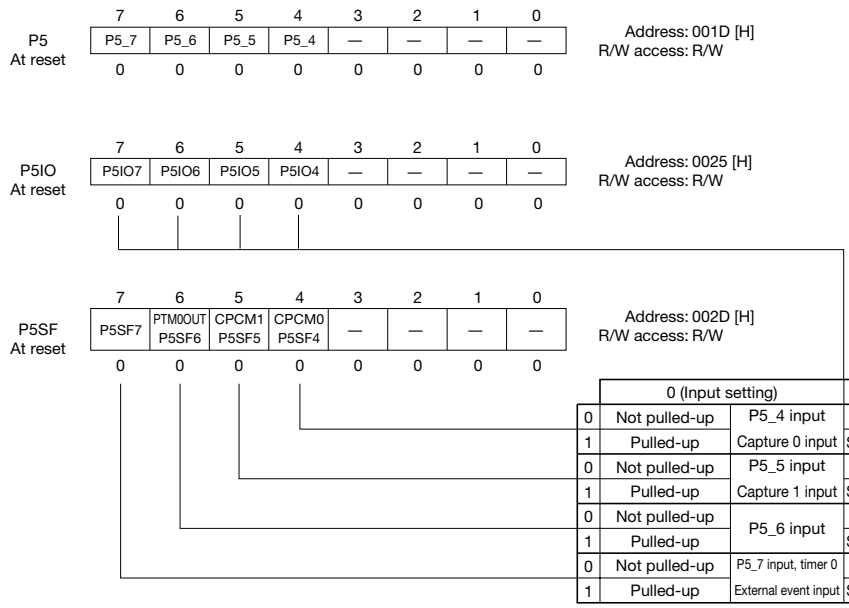
Port 5 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 5 mode register (P5IO). When output is specified (corresponding bits of P5IO = "1"), the value of the corresponding bits in the port 5 data register (P5) will be output from their appropriate pins.

In addition to its port function, P5 is assigned secondary functions (such as capture/compare I/O). If a secondary function output is to be used, set the corresponding bits of the port 5 mode register (P5IO) and the port 5 secondary function control register (P5SF) to "1". If a secondary function input is to be used, reset the corresponding bits of the port 5 mode register (P5IO) to "0" to configure the input mode (same input as the primary function input).

If the port is specified as an input (corresponding bits of P5IO = "0") and the port 5 secondary function control register (P5SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bit 7 of port 5 is set to secondary function output (P5IO7 = 1, P5SF7 = 1), the output will be fixed at "0", regardless of the value of the port 5 data register.

Figure 5-12 shows the configuration of the port 5 data register (P5), port 5 mode register (P5IO) and the port 5 secondary function control register (P5SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P5_4 input	Primary function	P5_4 output
1	Pulled-up	Capture 0 input	Secondary function	Compare 0 output
0	Not pulled-up	P5_5 input	Primary function	P5_5 output
1	Pulled-up	Capture 1 input	Secondary function	Compare 1 output
0	Not pulled-up	P5_6 input	Primary function	P5_6 output
1	Pulled-up		Secondary function	Timer 0 output
0	Not pulled-up	P5_7 input, timer 0	Primary function	P5_7 output
1	Pulled-up	External event input	Secondary function	0 output*

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-12 P5, P5IO, P5SF Configuration

Table 5-10 lists the data that is read, depending on the settings of P5IO and P5SF, when executing an instruction to read P5.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P5 will become a high impedance input port (P5IO = 00H, P5SF = 00H) and the contents of P5 will be 00H.

Table 5-10 P5 Read Data

	P5IO	P5SF	Read data
P5_4	0	*	P5_4/CPCM0 pin state
	1	0	Value of bit 4 of P5 (port data register)
	1	1	CPCM0 output data
P5_5	0	*	P5_5/CPCM1 pin state
	1	0	Value of bit 5 of P5 (port data register)
	1	1	CPCM1 output data
P5_6	0	*	P5_6 pin state
	1	0	Value of bit 6 of P5 (port data register)
	1	1	TM0OUT output data
P5_7	0	*	P5_7/TM0EVT pin state
	1	0	Value of bit 7 of P5 (port data register)
	1	1	"0"

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P5, depending on the settings of P5IO and P5SF, values will be read as listed in Table 5-10. The modified values will be written to P5 (port 5 data register).

5.10 Port 6 (P6)

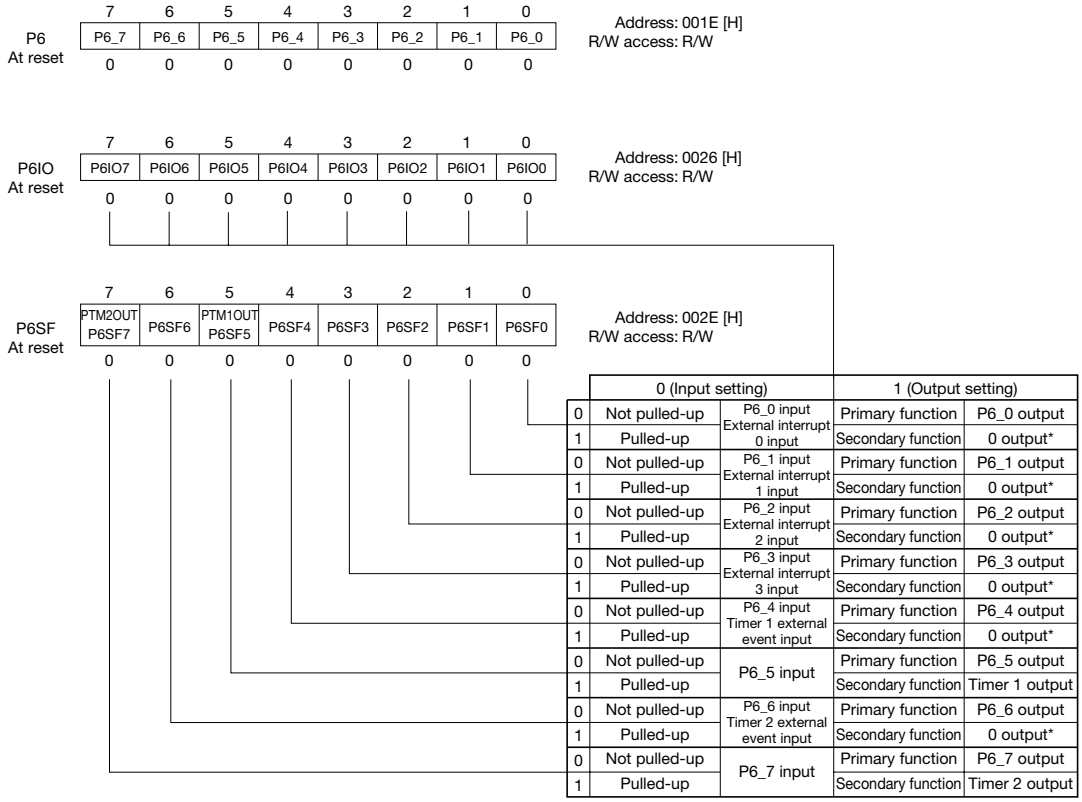
Port 6 is an 8-bit I/O port. Each individual bit can be specified as input or output by the port 6 mode register (P6IO). When output is specified (corresponding bits of P6IO = "1"), the value of the corresponding bits in the port 6 data register (P6) will be output from their appropriate pins.

In addition to its port function, P6 is assigned a secondary function (such as external interrupt input). If the secondary function output is to be used, set the corresponding bits of the port 6 mode register (P6IO) and the port 6 secondary function control register (P6SF) to "1". If the secondary function input is to be used, reset the corresponding bits of the port 6 mode register (P6IO) to "0" to configure the input mode (same input as the primary function input).

If the port is set as an input (corresponding bits of P6IO = "0") and the port 6 secondary function control register (P6SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bits 0 to 4 and bit 6 of port 6 are set as a secondary function output (P6IOn = 1, P6SFn = 1), the output will be fixed at "0", regardless of the value of the port 6 data register.

Figure 5-13 shows the configuration of the port 6 data register (P6), port 6 mode register (P6IO) and the port 6 secondary function control register (P6SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P6_0 input External interrupt 0 input	Primary function	P6_0 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_1 input External interrupt 1 input	Primary function	P6_1 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_2 input External interrupt 2 input	Primary function	P6_2 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_3 input External interrupt 3 input	Primary function	P6_3 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_4 input Timer 1 external event input	Primary function	P6_4 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_5 input	Primary function	P6_5 output
1	Pulled-up		Secondary function	Timer 1 output
0	Not pulled-up	P6_6 input Timer 2 external event input	Primary function	P6_6 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P6_7 input	Primary function	P6_7 output
1	Pulled-up		Secondary function	Timer 2 output

0 output*: "0" is output, regardless of the value of the port data register

Figure 5-13 P6, P6IO, P6SF Configuration

Table 5-11 lists the data that is read, depending on the settings of P6IO and P6SF, when executing an instruction to read P6.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P6 will become a high impedance input port (P6IO = 00H, P6SF = 00H) and the contents of P6 will be 00H.

Table 5-11 P6 Read Data

	P6IO	P6SF	Read data
P6_0	0	*	P6_0/EXINT0 pin state
	1	0	Value of bit 0 of P6 (port data register)
	1	1	"0"
P6_1	0	*	P6_1/EXINT1 pin state
	1	0	Value of bit 1 of P6 (port data register)
	1	1	"0"
P6_2	0	*	P6_2/EXINT2 pin state
	1	0	Value of bit 2 of P6 (port data register)
	1	1	"0"
P6_3	0	*	P6_3/EXINT3 pin state
	1	0	Value of bit 3 of P6 (port data register)
	1	1	"0"
P6_4	0	*	P6_4/TM1EVT pin state
	1	0	Value of bit 4 of P6 (port data register)
	1	1	"0"
P6_5	0	*	P6_5 pin state
	1	0	Value of bit 5 of P6 (port data register)
	1	1	TM1OUT output data
P6_6	0	*	P6_6 pin state
	1	0	Value of bit 6 of P6 (port data register)
	1	1	"0"
P6_7	0	*	P6_7 pin state
	1	0	Value of bit 7 of P6 (port data register)
	1	1	TM2OUT output data

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P6, depending on the settings of P6IO and P6SF, values will be read as listed in Table 5-11. The modified values will be written to P6 (port 6 data register).

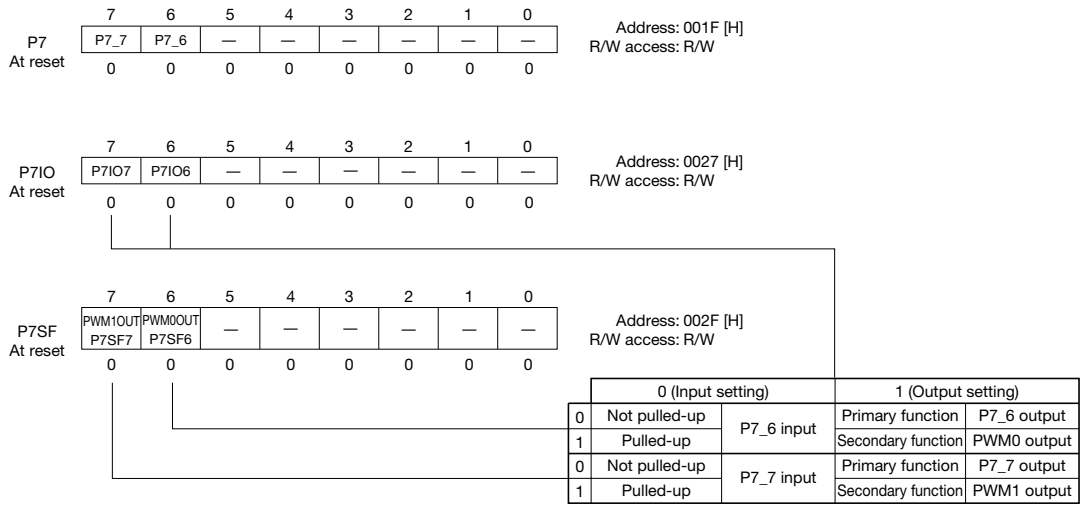
5.11 Port 7 (P7)

Port 7 is a 7-bit I/O port. Each individual bit can be specified as input or output by the port 7 mode register (P7IO). When output is specified (corresponding bits of P7IO = "1"), the value of the corresponding bits in the port 7 data register (P7) will be output from their appropriate pins.

In addition to its port function, P7 is assigned secondary functions (such as PWM0 output). If a secondary function output is to be used, set the corresponding bits of the port 7 mode register (P7IO) and the port 7 secondary function control register (P7SF) to "1".

If the port is set as an input (corresponding bits of P7IO = "0") and the port 7 secondary function control register (P7SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-14 shows the configuration of the port 7 data register (P7), port 7 mode register (P7IO) and the port 7 secondary function control register (P7SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P7_6 input	Primary function	P7_6 output
1	Pulled-up		Secondary function	PWM0 output
0	Not pulled-up	P7_7 input	Primary function	P7_7 output
1	Pulled-up		Secondary function	PWM1 output

"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-14 P7, P7IO, P7SF Configuration

Table 5-12 lists the data that is read, depending on the settings of P7IO and P7SF, when executing an instruction to read P7.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P7 will become a high impedance input port (P7IO = 00H, P7SF = 00H) and the contents of P7 will be 00H.

Table 5-12 P7 Read Data

	P7IO	P7SF	Read data
P7_6	0	*	P7_6 pin state
	1	0	Value of bit 6 of P7 (port data register)
	1	1	PWM0OUT output data
P7_7	0	*	P7_7 pin state
	1	0	Value of bit 7 of P7 (port data register)
	1	1	PWM1OUT output data

"*" indicates "0" or "1"

[Note]

If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P7, depending on the settings of P7IO and P7SF, values will be read as listed in Table 5-12. The modified values will be written to P7 (port 7 data register).

5.12 Port 8 (P8)

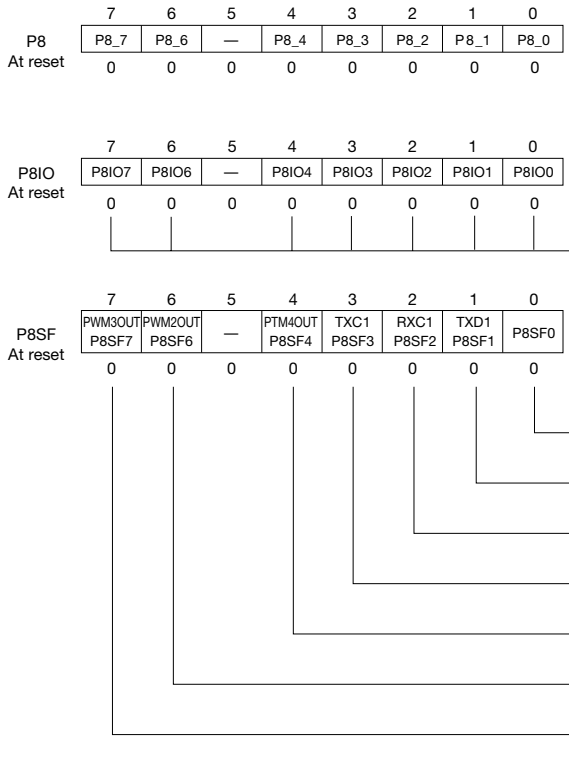
Port 8 is a 7-bit I/O port. Each individual bit can be specified as input or output by the port 8 mode register (P8IO). When output is specified (corresponding bits of P8IO = "1"), the value of the corresponding bits in the port 8 data register (P8) will be output from their appropriate pins.

In addition to its port function, P8 is assigned secondary functions (such as SIO1 receive data input). If a secondary function output is to be used, set the corresponding bits of the port 8 mode register (P8IO) and the port 8 secondary function control register (P8SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 8 mode register (P8IO) to "0" to configure the input mode (same input as the primary function input).

If the port is set as an input (corresponding bits of P8IO = "0") and the port 8 secondary function control register (P8SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bit 0 of port 8 is set as a secondary function output (P8IO0 = 1, P8SF0 = 1), the output will be fixed at "0", regardless of the value of the port 8 data register.

Figure 5-15 shows the configuration of the port 8 data register (P8), port 8 mode register (P8IO) and the port 8 secondary function control register (P8SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P8_0 input SIO1 receive data input	Primary function	P8_0 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P8_1 input	Primary function	P8_1 output
1	Pulled-up		Secondary function	SIO1 transmit data output
0	Not pulled-up	P8_2 input SIO1 receive clock input	Primary function	P8_2 output
1	Pulled-up		Secondary function	SIO1 receive clock output
0	Not pulled-up	P8_3 input SIO1 transmit clock input	Primary function	P8_3 output
1	Pulled-up		Secondary function	SIO1 transmit clock output
0	Not pulled-up	P8_4 input	Primary function	P8_4 output
1	Pulled-up		Secondary function	Timer 4 output
0	Not pulled-up	P8_6 input	Primary function	P8_6 output
1	Pulled-up		Secondary function	PWM2 output
0	Not pulled-up	P8_7 input	Primary function	P8_7 output
1	Pulled-up		Secondary function	PWM3 output

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-15 P8, P8IO, P8SF Configuration

Table 5-13 lists the data that is read, depending on the settings of P8IO and P8SF, when executing an instruction to read P8.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P8 will become a high impedance input port (P8IO = 00H, P8SF = 00H) and the contents of P8 will be 00H.

Table 5-13 P8 Read Data

	P8IO	P8SF	Read data
P8_0	0	*	P8_0/RXD1 pin state
	1	0	Value of bit 0 of P8 (port data register)
	1	1	"0"
P8_1	0	*	P8_1 pin state
	1	0	Value of bit 1 of P8 (port data register)
	1	1	TXD1 output data
P8_2	0	*	P8_2/RXC1 pin state
	1	0	Value of bit 2 of P8 (port data register)
	1	1	RXC1 output data
P8_3	0	*	P8_3/TXC1 pin state
	1	0	Value of bit 3 of P8 (port data register)
	1	1	TXC1 output data
P8_4	0	*	P8_4 pin state
	1	0	Value of bit 4 of P8 (port data register)
	1	1	TM4OUT output data
P8_6	0	*	P8_6 pin state
	1	0	Value of bit 6 of P8 (port data register)
	1	1	PWM2OUT output data
P8_7	0	*	P8_7 pin state
	1	0	Value of bit 7 of P8 (port data register)
	1	1	PWM3OUT output data

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P8, depending on the settings of P8IO and P8SF, values will be read as listed in Table 5-13. The modified values will be written to P8 (port 8 data register).

5.13 Port 9 (P9)

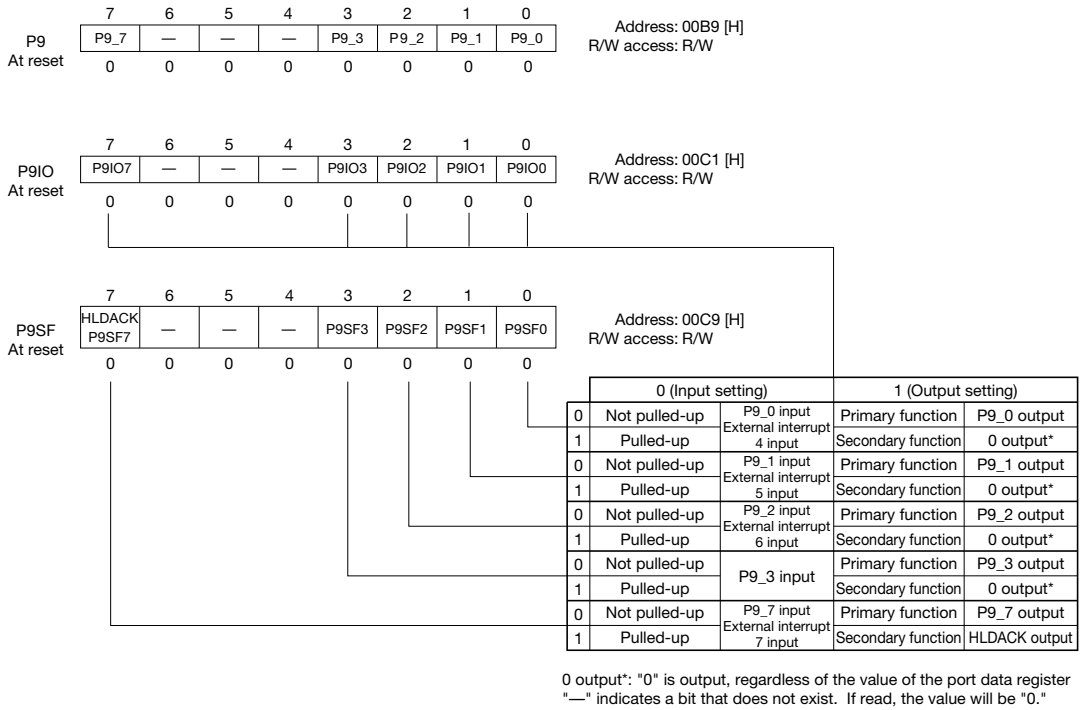
Port 9 is a 5-bit I/O port. Each individual bit can be specified as input or output by the port 9 mode register (P9IO). When output is specified (corresponding bits of P9IO = "1"), the value of the corresponding bits in the port 9 data register (P9) will be output from their appropriate pins.

In addition to its port function, P9 is assigned secondary functions (such as external interrupt input). If a secondary function output is to be used, set the corresponding bits of the port 9 mode register (P9IO) and the port 9 secondary function control register (P9SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 9 mode register (P9IO) to "0" to configure the input mode (same input as the primary function input).

If the port is set as an input (corresponding bits of P9IO = "0") and the port 9 secondary function control register (P9SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bits 0 to 3 of port 9 are set as secondary function outputs (P9IOn = 1, P9SFn = 1), the output will be fixed at "0", regardless of the value of the port 9 data register.

Figure 5-16 shows the configuration of the port 9 data register (P9), port 9 mode register (P9IO) and the port 9 secondary function control register (P9SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P9_0 input External interrupt 4 input	Primary function	P9_0 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P9_1 input External interrupt 5 input	Primary function	P9_1 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P9_2 input External interrupt 6 input	Primary function	P9_2 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P9_3 input	Primary function	P9_3 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P9_7 input External interrupt 7 input	Primary function	P9_7 output
1	Pulled-up		Secondary function	HLDACK output

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-16 P9, P9IO, P9SF Configuration

Table 5-14 lists the data that is read, depending on the settings of P9IO and P9SF, when executing an instruction to read P9.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P9 will become a high impedance input port (P9IO = 00H, P9SF = 00H) and the contents of P9 will be 00H.

Table 5-14 P9 Read Data

	P9IO	P9SF	Read data
P9_0	0	*	P9_0/EXINT4 pin state
	1	0	Value of bit 0 of P9 (port data register)
	1	1	"0"
P9_1	0	*	P9_1/EXINT5 pin state
	1	0	Value of bit 1 of P9 (port data register)
	1	1	"0"
P9_2	0	*	P9_2/EXINT6 pin state
	1	0	Value of bit 2 of P9 (port data register)
	1	1	"0"
P9_3	0	*	P9_3/EXINT7 pin state
	1	0	Value of bit 3 of P9 (port data register)
	1	1	"0"
P9_7	0	*	P9_7 pin state
	1	0	Value of bit 7 of P9 (port data register)
	1	1	HLDACK output

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P9, depending on the settings of P9IO and P9SF, values will be read as listed in Table 5-14. The modified values will be written to P9 (port 9 data register).

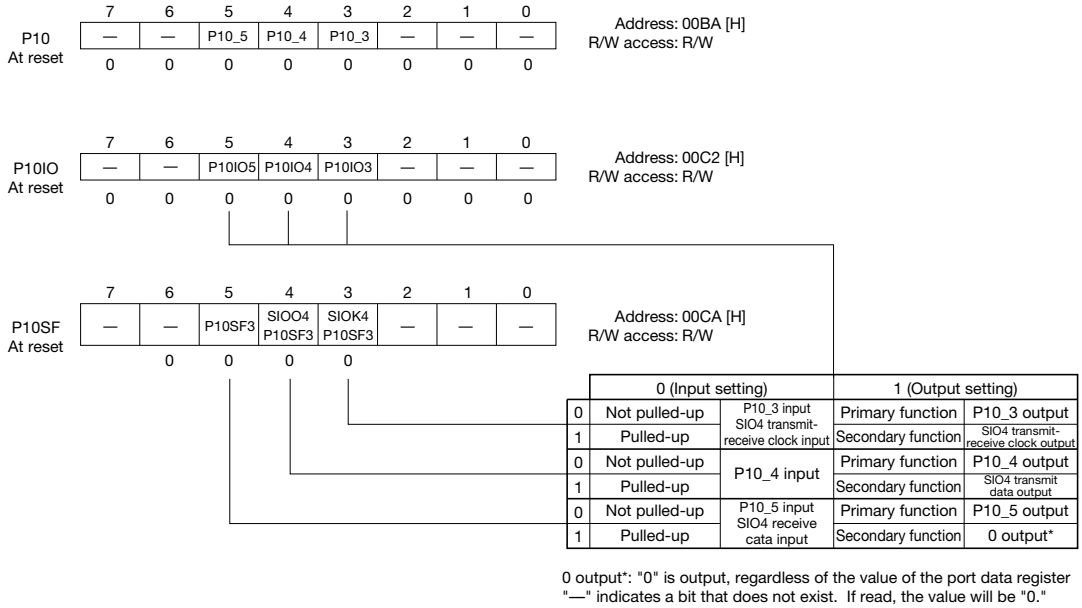
5.14 Port 10 (P10)

Port 10 is a 3-bit I/O port. Each individual bit can be specified as input or output by the port 10 mode register (P10IO). When output is specified (corresponding bits of P10IO = "1"), the value of the corresponding bits in the port 10 data register (P10) will be output from their appropriate pins.

In addition to its port function, P10 is assigned secondary functions (such as SIO4 transmit-receive clock I/O). If a secondary function output is to be used, set the corresponding bits of the port 10 mode register (P10IO) and the port 10 secondary function control register (P10SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 10 mode register (P10IO) to "0" to configure the input mode (same input as the primary function input).

If the port is set as an input (corresponding bits of P10IO = "0") and the port 10 secondary function control register (P10SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bit 5 of port 10 is set as secondary function outputs (P10IOn = 1, P10SFn = 1), the output will be fixed at "0", regardless of the value of the port 10 data register.
Figure 5-17 shows the configuration of the port 10 data register (P10), port 10 mode register (P10IO) and the port 10 secondary function control register (P10SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P10_3 input SIO4 transmit-receive clock input	Primary function	P10_3 output
1	Pulled-up		Secondary function	SIO4 transmit-receive clock output
0	Not pulled-up	P10_4 input	Primary function	P10_4 output
1	Pulled-up		Secondary function	SIO4 transmit data output
0	Not pulled-up	P10_5 input SIO4 receive cata input	Primary function	P10_5 output
1	Pulled-up		Secondary function	0 output*

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-17 P10, P10IO, P10SF Configuration

Table 5-15 lists the data that is read, depending on the settings of P10IO and P10SF, when executing an instruction to read P10.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P10 will become a high impedance input port (P10IO = 00H, P10SF = 00H) and the contents of P10 will be 00H.

Table 5-15 P10 Read Data

	P10IO	P10SF	Read data
P10_3	0	*	P10_3/SIOCK4 pin state
	1	0	Value of bit 3 of P10 (port data register)
	1	1	SIOCK4 output data
P10_4	0	*	P10_4 pin state
	1	0	Value of bit 4 of P10 (port data register)
	1	1	SIOO4 output data
P10_5	0	*	P10_5/SIO14 pin state
	1	0	Value of bit 5 of P10 (port data register)
	1	1	"0"

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P10, depending on the settings of P10IO and P10SF, values will be read as listed in Table 5-15. The modified values will be written to P10 (port 10 data register).

5.15 Port 11 (P11)

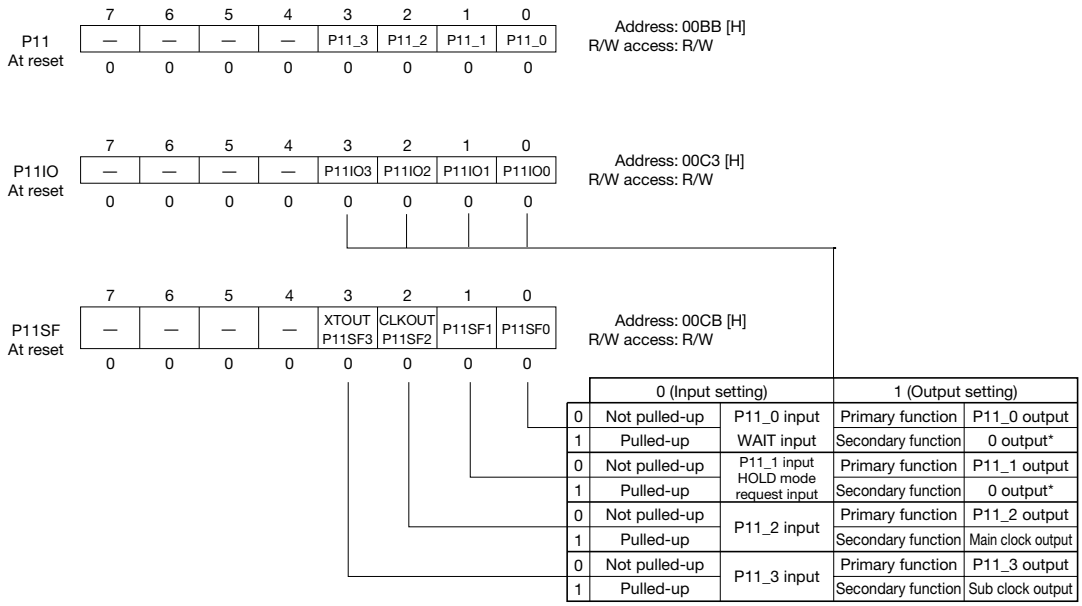
Port 11 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 11 mode register (P11IO). When output is specified (corresponding bits of P11IO = "1"), the value of the corresponding bits in the port 11 data register (P11) will be output from their appropriate pins.

In addition to its port function, P11 is assigned secondary functions (such as WAIT input). If a secondary function output is to be used, set the corresponding bits of the port 11 mode register (P11IO) and the port 11 secondary function control register (P11SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 11 mode register (P11IO) to "0" to configure the input mode (same input as the primary function input).

If the port is set as an input (corresponding bits of P11IO = "0") and the port 11 secondary function control register (P11SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bits 0 and 1 of port 11 are set as secondary function outputs (P11IOn = 1, P11SFn = 1), the output will be fixed at "0", regardless of the value of the port 11 data register.

Figure 5-18 shows the configuration of the port 11 data register (P11), port 11 mode register (P11IO) and the port 11 secondary function control register (P11SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P11_0 input	Primary function	P11_0 output
1	Pulled-up	WAIT input	Secondary function	0 output*
0	Not pulled-up	P11_1 input HOLD mode request input	Primary function	P11_1 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P11_2 input	Primary function	P11_2 output
1	Pulled-up		Secondary function	Main clock output
0	Not pulled-up	P11_3 input	Primary function	P11_3 output
1	Pulled-up		Secondary function	Sub clock output

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-18 P11, P11IO, P11SF Configuration

Table 5-16 lists the data that is read, depending on the settings of P11IO and P11SF, when executing an instruction to read P11.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P11 will become a high impedance input port (P11IO = 00H, P11SF = 00H) and the contents of P11 will be 00H.

Table 5-16 P11 Read Data

	P11IO	P11SF	Read data
P11_0	0	*	P11_0/WAIT pin state
	1	0	Value of bit 0 of P11 (port data register)
	1	1	"0"
P11_1	0	*	P11_1/HOLD pin state
	1	0	Value of bit 1 of P11 (port data register)
	1	1	"0"
P11_2	0	*	P11_2 pin state
	1	0	Value of bit 2 of P11 (port data register)
	1	1	CLKOUT output data
P11_3	0	*	P11_3 pin state
	1	0	Value of bit 3 of P11 (port data register)
	1	1	XTOUT output data

"*" indicates "0" or "1"

[Note]
If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P11, depending on the settings of P11IO and P11SF, values will be read as listed in Table 5-16. The modified values will be written to P11 (port 11 data register).

5.16 Port 12 (P12)

Port 12 is an 8-bit input-only port. Therefore, there is no mode register or secondary function control register.

The pin status can be read by the port 12 data register (P12).

In addition to its port function, a secondary function (analog input for A/D converter) is assigned to P12 (same input as the primary function input).

There are no pulled-up inputs at port 12.

Figure 5-19 shows the configuration of the port 12 data register (P12). Table 5-17 lists the P12 read data.

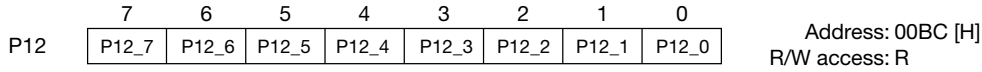


Figure 5-19 P12 Configuration

Table 5-17 P12 Read Data

	Read data
P12_0	P12_0/AI0 pin state
P12_1	P12_1/AI1 pin state
P12_2	P12_2/AI2 pin state
P12_3	P12_3/AI3 pin state
P12_4	P12_4/AI4 pin state
P12_5	P12_5/AI5 pin state
P12_6	P12_6/AI6 pin state
P12_7	P12_7/AI7 pin state

5.17 Port 14 (P14)

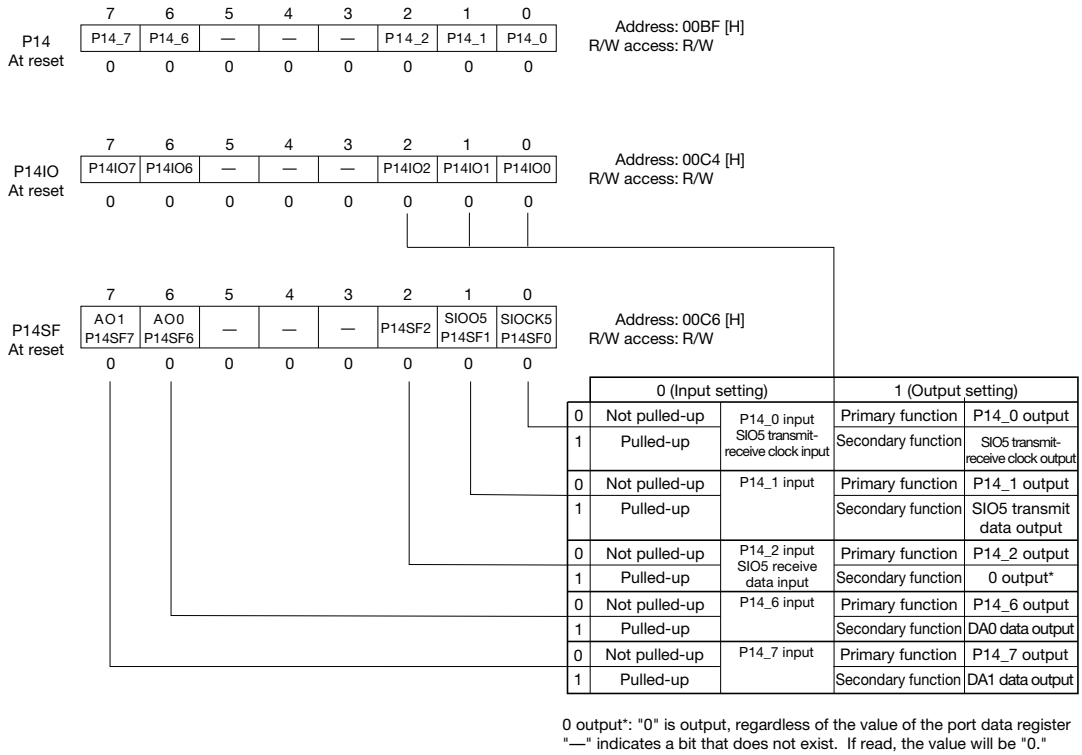
Port 14 is a 5-bit I/O port. Each individual bit can be specified as input or output by the port 14 mode register (P14IO). When output is specified (corresponding bits of P14IO = "1"), the value of the corresponding bits in the port 14 data register (P14) will be output from their appropriate pins.

In addition to its port function, P14 is assigned secondary functions (such as SIO5 receive data input). If a secondary function output is to be used, set the corresponding bits of the port 14 mode register (P14IO) and the port 14 secondary function control register (P14SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 14 mode register (P14IO) to "0" to configure the input mode (same input as the primary function input).

If the port is configured as an input (corresponding bits of P14IO = "0") and the port 14 secondary function control register (P14SF) is set to "1", inputs will be pulled-up at the pins corresponding to those bits.

If bit 2 of port 14 is configured as a secondary function output (P14IO0 = 1, P14SF0 = 1), the output will be fixed at "0", regardless of the value of the port 14 data register.

Figure 5-20 shows the configuration of the port 14 data register (P14), port 14 mode register (P14IO) and the port 14 secondary function control register (P14SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P14_0 input SIO5 transmit-receive clock input	Primary function	P14_0 output
1	Pulled-up		Secondary function	SIO5 transmit-receive clock output
0	Not pulled-up	P14_1 input	Primary function	P14_1 output
1	Pulled-up		Secondary function	SIO5 transmit data output
0	Not pulled-up	P14_2 input SIO5 receive data input	Primary function	P14_2 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P14_6 input	Primary function	P14_6 output
1	Pulled-up		Secondary function	DA0 data output
0	Not pulled-up	P14_7 input	Primary function	P14_7 output
1	Pulled-up		Secondary function	DA1 data output

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-20 P14, P14IO, P14SF Configuration

Table 5-18 lists the data that is read, depending on the settings of P14IO and P14SF, when executing an instruction to read P14.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P14 will become a high impedance input port (P14IO = 00H, P14SF = 00H) and the contents of P14 will be 00H.

Table 5-18 P14 Read Data

	P14IO	P14SF	Read data
P14_0	0	*	P14_0/SIOCK5 pin state
	1	0	Value of bit 0 of P14 (port data register)
	1	1	SIOCK5 output data
P14_1	0	*	P14_1 pin state
	1	0	Value of bit 1 of P14 (port data register)
	1	1	SIOO5 output data
P14_2	0	*	P14_2/SIOI5 pin state
	1	0	Value of bit 2 of P14 (port data register)
	1	1	"0"
P14_6	0	*	P14_6 pin state
	1	0	Value of bit 6 of P14 (port data register)
	1	1	AO0 output data
P14_7	0	*	P14_7 pin state
	1	0	Value of bit 7 of P14 (port data register)
	1	1	AO1 output data

"*" indicates "0" or "1"

[Note]
If arthmetic, SB, RB, XORB or other read-modify-write instructions are executed for P14, depending on the settings of P14IO and P14SF, values will be read as listed in Table 5-18. The modified values will be written to P14 (port 14 data register).

5.18 Port 15 (P15)

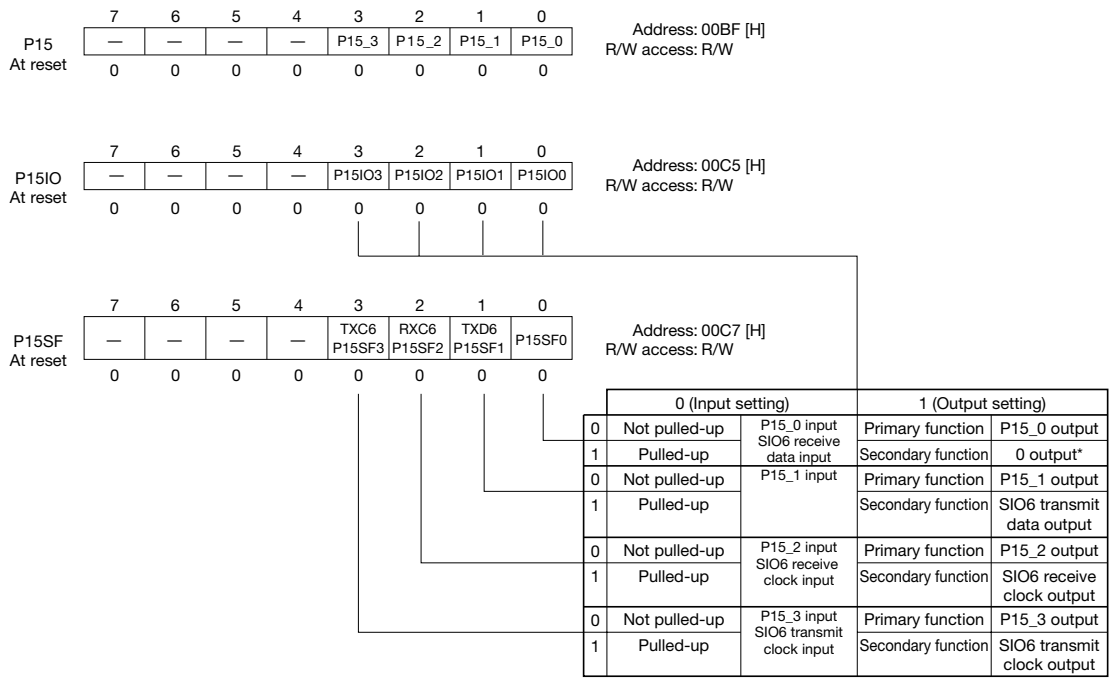
Port 15 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 15 mode register (P15IO). When output is specified (corresponding bits of P15IO = "1"), the value of the corresponding bits in the port 15 data register (P15) will be output from their appropriate pins.

In addition to its port function, P15 is assigned secondary functions (such as SIO6 receive data input). If a secondary function output is to be used, set the corresponding bits of the port 15 mode register (P15IO) and the port 15 secondary function control register (P15SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 15 mode register (P15IO) to "0" to configure the input mode (same input as the primary function input).

If the port is configured as an input (corresponding bits of P15IO = "0") and the port 15 secondary function control register (P15SF) is set to "1", inputs will be pulled-up at the pins corresponding to those bits.

If bit 0 of port 15 is configured as a secondary function output (P15IO0 = 1, P15SF0 = 1), the output will be fixed at "0", regardless of the value of the port 15 data register.

Figure 5-21 shows the configuration of the port 15 data register (P15), port 15 mode register (P15IO) and the port 15 secondary function control register (P15SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P15_0 input SIO6 receive data input	Primary function	P15_0 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P15_1 input	Primary function	P15_1 output
1	Pulled-up		Secondary function	SIO6 transmit data output
0	Not pulled-up	P15_2 input SIO6 receive clock input	Primary function	P15_2 output
1	Pulled-up		Secondary function	SIO6 receive clock output
0	Not pulled-up	P15_3 input SIO6 transmit clock input	Primary function	P15_3 output
1	Pulled-up		Secondary function	SIO6 transmit clock output

0 output*: "0" is output, regardless of the value of the port data register
"—" indicates a bit that does not exist. If read, the value will be "0."

Figure 5-21 P15, P15IO, P15SF Configuration

Table 5-19 lists the data that is read, depending on the settings of P15IO and P15SF, when executing an instruction to read P15.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), P15 will become a high impedance input port (P15IO = 00H, P15SF = 00H) and the contents of P15 will be 00H.

Table 5-19 P15 Read Data

	P15IO	P15SF	Read data
P15_0	0	*	P15_0/RXD6 pin state
	1	0	Value of bit 0 of P15 (port data register)
	1	1	"0"
P15_1	0	*	P15_1 pin state
	1	0	Value of bit 1 of P15 (port data register)
	1	1	TXD6 output data
P15_2	0	*	P15_2/RXC6 pin state
	1	0	Value of bit 2 of P15 (port data register)
	1	1	RXC6 output data
P15_3	0	*	P15_3/TXC6 pin state
	1	0	Value of bit 3 of P15 (port data register)
	1	1	TXC6 output data

"*" indicates "0" or "1"

[Note]
If arthmetic, SB, RB, XORB or other read-modify-write instructions are executed for P15, depending on the settings of P15IO and P15SF, values will be read as listed in Table 5-19. The modified values will be written to P15 (port 15 data register).

Chapter 6

Clock Oscillation Circuit

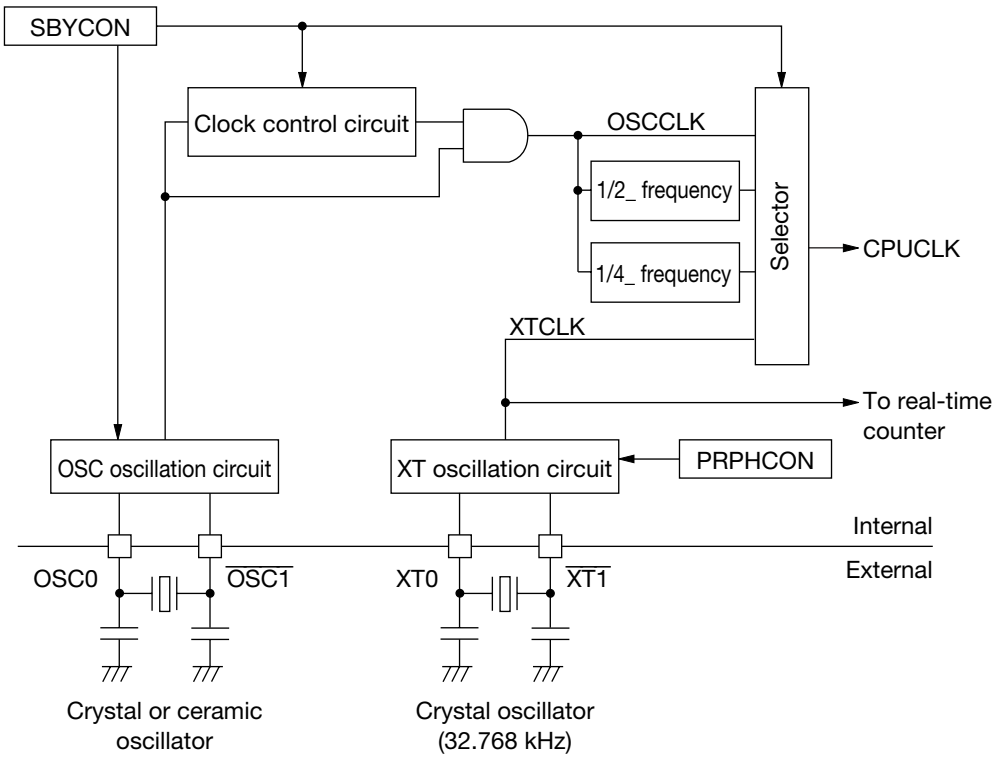
6. Clock Oscillation Circuit

6.1 Overview

The MSM66577 family has two systems of internal clock oscillation circuits, OSC and XT. Four types of clocks can be selected as the CPU operating clock (CPUCLK): OSCCLK (main clock), frequency divided clocks (1/2OSCCLK, 1/4OSCCLK), and XTCLK (sub clock). The current consumed during operation can be reduced by changing the clock in response to the operating conditions. XT is used mainly for the real-time counter. Externally generated clocks can be input directly to both OSC and XT.

6.2 Clock Oscillation Circuit Configuration

Figure 6-1 shows the configuration of the clock oscillation circuit.



OSCCLK: Main clock
XTCLK: Subclock (32.768 kHz)
CPUCLK: CPU operating clock
SBYCON: Standby control register
PRPHCON: Peripheral control register

Figure 6-1 Clock Oscillation Circuit Configuration

6.3 Clock Oscillation Circuit Registers

Table 6-1 lists a summary of the SFRs for clock oscillation circuit control.

Table 6-1 Summary of SFRs for Clock Oscillation Circuit Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
000F	Standby control register	SBYCON	—	R/W	8	08	3-4
0015☆	Peripheral control register	PRPHCON	—	R/W	8	8C	15-2

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

6.4 OSC Oscillation Circuit

The OSC oscillation circuit generates the main clock pulse (OSCCLK). A crystal oscillator and other required elements are connected to OSC0 and $\overline{OSC1}$.

Figure 6-2 shows the configuration of the OSC oscillation circuit. Figure 6-3 shows an example connection of an OSC crystal oscillation circuit.

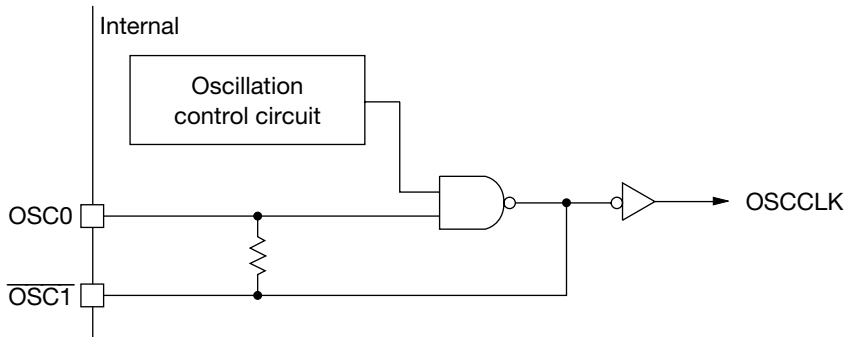


Figure 6-2 OSC Oscillation Circuit Configuration

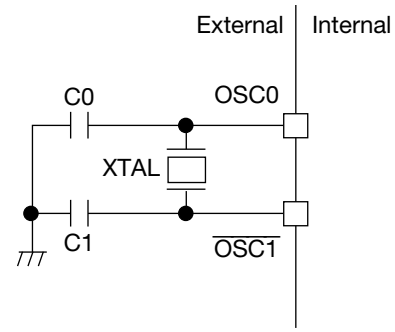


Figure 6-3 OSC Crystal Oscillation Circuit Connection Example

[Notes]

1. The values of C0 and C1 must be set based on the specifications of the external crystal (XTAL).
2. Instead of XTAL, a ceramic resonator may be used.
3. Depending upon the frequency band used, additional components (not shown) may be required.

If the main clock pulse (OSCCLK) is to be supplied externally, connect it directly to the OSC0 pin input. Leave the $\overline{OSC1}$ pin open (unconnected).

Figure 6-4 shows an example connection when the OSC clock is input externally.

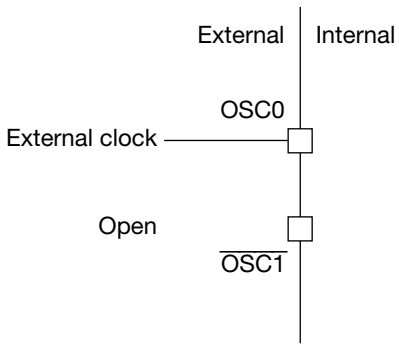


Figure 6-4 Connection Example for External OSC Clock Input

[Note]

If an external clock is to be used for operation, keep the clock pulse width as specified by the AC characteristics.

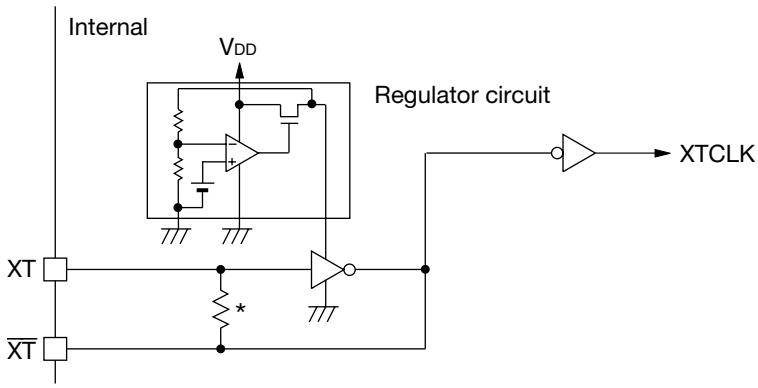
The standby control register (SBYCON) can be set to halt the OSC oscillation circuit. When resuming oscillation of the OSC oscillation circuit from a halted state, the main clock pulse (OSCCLK) will be transmit after waiting for the oscillation stabilization time, the number of clock cycles specified by OST0 and OST1 (bits 4 and 5) of SBYCON. Because the oscillation stabilization time differs depending upon the oscillator used, externally mounted components, and the frequency band, first verify the actual oscillation stabilization time of the circuit board in the product application, and then set SBYCON with the wait time until suitable oscillation stabilization is achieved.

6.5 XT Oscillation Circuit

The XT oscillation circuit generates the subclock pulse (XTCLK). A crystal oscillator of 32.768 kHz and other required elements are connected to XT0 and $\overline{XT1}$.

To reduce power consumption, the XT oscillation circuit operates at a voltage level that is regulated internally. Before an external clock is to be input, set EXTXT (bit 4) of the peripheral control register (PRPHCON) to "1". This switches the internally regulated voltage to V_{DD} and turns OFF the oscillation feedback resistors.

Figure 6-5 shows the configuration of the XT oscillation circuit. Figure 6-6 shows an example connection of the XT crystal oscillation circuit.



*If the EXTXT flag of PRPHCON is set to "1", feedback resistors are OFF.

Figure 6-5 XT Oscillation Circuit Configuration

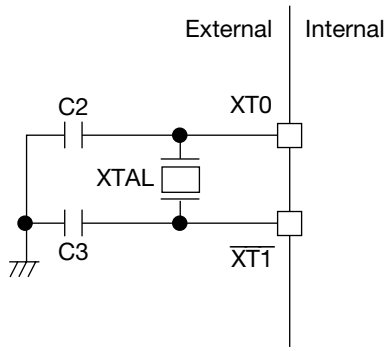


Figure 6-6 XT Crystal Oscillation Circuit Connection Example

[Notes]

1. The values of C2 and C3 must be set based on the specification of the external XTAL (32.768 kHz).
2. Because the XT oscillation circuit was designed to be connected to an extremely low power crystal, there may not be any oscillation if another type of crystal is connected.

If the subclock pulse (XTCLK) is to be supplied externally, connect it to the XT0 pin input and leave the $\overline{XT1}$ pin open (unconnected).

Figure 6-7 shows an example connection when the XT clock is input externally.

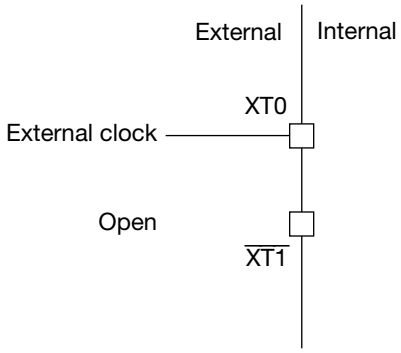


Figure 6-7 Connection Example for External XT Clock Input

[Note]
Before an external clock is to be used in the XT oscillation circuit, set EXTXT (bit 4) of PRPHCON to "1".

The XT oscillation circuit cannot be halted by the program. Because there is no circuit to control the oscillation stabilization time, from the time when power is turned on until overflow of the real-time counter causes bit 12 (RTC12) to be set, do not select the sub clock (XTCLK) as the CPU operation clock (CPUCLK).

If the sub clock (XTCLK) is not used, fix the XT0 pin at GND level and set EXTXT (bit 4) of PRPHCON to "1".

Chapter 7

Time Base Counter (TBC)

7. Time Base Counter (TBC)

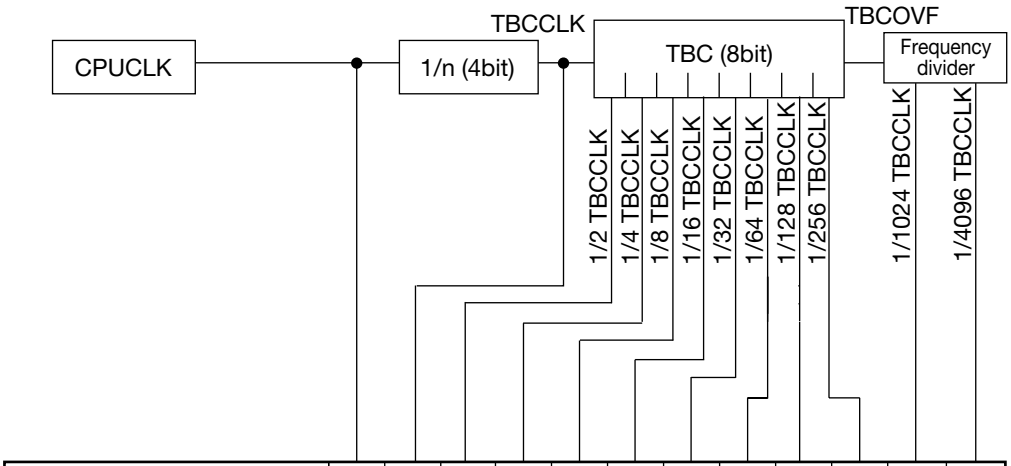
7.1 Overview

The MSM66577 family has an 8-bit internal time base counter (TBC) to generate a reference clock for internal peripheral modules.

The front stage of the TBC has an auto-reload type 4-bit 1/n counter. Base clocks can be generated for internal peripheral from the wide-ranging CPUCLK frequency.

7.2 Time Base Counter (TBC) Configuration

Figure 7-1 shows the TBC configuration.



PWM	√		√	√							
SIO4, 5				√	√		√				
16-bit FRC		√	√	√	√	√	√	√	√		
16-bit timer 0		√	√	√	√	√	√	√			
8-bit timer 1, 2		√	√	√	√		√			√	
8-bit timer 3/BRG		√	√	√	√		√			√	
8-bit timer 4/BRG		√	√	√	√		√			√	
8-bit timer 5/BRG		√	√	√	√		√			√	
8-bit timer 6/WDT					√	√	√	√	√	√	√	√
8-bit timer 9		√	√	√	√		√			√		

Figure 7-1 TBC Configuration

7.3 Time Base Counter Registers

Table 7-1 lists a summary of SFRs for time base counter control.

Table 7-1 Summary of SFRs for Time Base Counter Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0060☆	TBC clock divider register	TBCKDVR	TBCKDV	R/W	8/16	F0	7-3
0061☆	TBC clock divider counter	—		R	16	F0	7-2

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

7.4 1/n Counter

To generate base clocks for internal peripheral modules from the wide ranging CPUCLK frequency, the MSM66577 family is equipped with a 4-bit auto-reload timer into which CPUCLK is input.
This 1/n counter consists of a 4-bit counter (TBC clock dividing counter) and a 4-bit register that stores the reload value (TBC clock divider register).

7.4.1 Description of 1/n Counter Registers

(1) TBC clock dividing counter (TBCKDV upper 8 bits)

The TBC clock dividing counter (upper 8 bits of TBCKDV) is a 4-bit counter and its input is CPUCLK. When the counter overflows it is loaded with the contents of the TBC clock divider register (TBCKDVR).

The TBC clock dividing counter (upper 8 bits of TBCKDV) can be accessed only in word sized units. The value of the TBC clock dividing counter is read from the four bits of bit 8 through bit 11. If the upper 4 bits are read, a value of "1" will always be obtained. The TBC clock divider register (TBCKDVR) is read from the lower 8 bits of TBCKDV.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the upper 8 bits of TBCKDV become F0H.

Figure 7-2 shows the configuration of the upper 8 bits of TBCKDV.

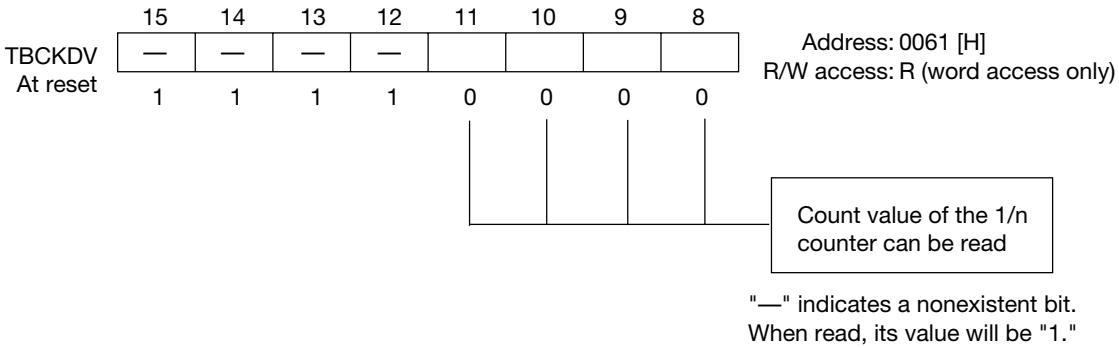


Figure 7-2 Configuration of Upper 8 Bits of TBCKDV

(2) TBC clock divider register (TBCKDVR)

The TBC clock divider register (TBCKDVR) consists of 4 bits. This register stores the value to be reloaded into the TBC clock dividing counter.

TBCKDVR can be read from or written to by the program. However, write operations are not valid for bits 4 through 7. If read, bits 4 through 7 are always "1".

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TBCKDVR becomes F0H.

[Note]
When reset, the 1/n counter divides CPUCLK by 16 and 1/16CPUCLK is supplied to TBC as TBCCLK. Therefore, after writing a reload value to TBCKDVR, there may be at most a delay of 16 CPUCLK pulses before the start of the division operation (as per the written value).

Figure 7-3 shows the configuration of TBCKDVR. Table 7-2 lists the correspondence between TBCKDVR settings and TBCCLK.

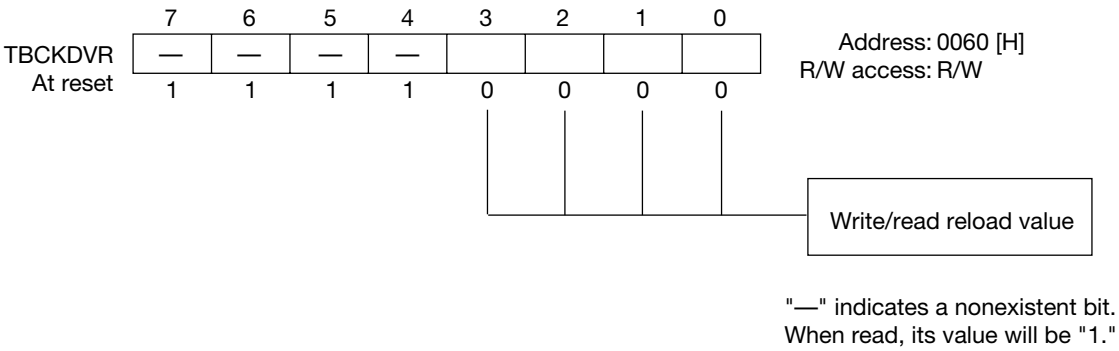


Figure 7-3 TBCKDVR (Lower 8 Bits of TBCKDV) Configuration

Table 7-2 Correspondence between TBCKDVR Settings and TBCCLK

Value of TBCKDVR settings [H]	TBCCLK
F0	1/16 CPUCLK
F1	1/15 CPUCLK
F2	1/14 CPUCLK
F3	1/13 CPUCLK
F4	1/12 CPUCLK
F5	1/11 CPUCLK
F6	1/10 CPUCLK
F7	1/9 CPUCLK
F8	1/8 CPUCLK
F9	1/7 CPUCLK
FA	1/6 CPUCLK
FB	1/5 CPUCLK
FC	1/4 CPUCLK
FD	1/3 CPUCLK
FE	1/2 CPUCLK
FF	1/1 CPUCLK

7.4.2 Example of 1/n Counter-related Register Settings

- TBC clock divider register (TBCKDVR)
  This register stores the reload value to the TBC clock dividing counter. When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the reload value becomes F0H, and TBCCLK becomes CPUCLK divided by 16 (1/16CPUCLK). If TBCCLK is set to 1/1CPUCLK, the reload value becomes FFH.

7.5 Time Base Counter (TBC) Operation

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the time base counter (TBC) is reset to "0". Thereafter, as long as the original oscillation (CPCLK) supply is not halted, operation will continue by TBCCLK that has been divided by the front stage 1/n counter.

Overflow of TBC is divided further by a frequency divider circuit, and supplied to the general-purpose 8-bit timer 6 (that also functions as the watchdog timer).

Chapter 8

General-Purpose 8/16 Bit Timers

8. General-Purpose 8/16 Bit Timers

8.1 Overview

The MSM66577 family has the following internal general-purpose timers: a 16-bit auto-reload timer (timer 0), six 8-bit auto-reload timers (timers 1, 2, 3, 4, 5, and 9), and an 8-bit auto-reload timer that also functions as a watchdog timer (timer 6). Timers 1 and 2 can be combined and used as a 16-bit timer.

8.2 General-purpose 8-bit/16-bit Timer Configurations

Table 8-1 lists a summary of the function of each general-purpose timer. Marks (√) within the table indicate that a function can be selected. Dashes (—) indicate that the function cannot be selected.

Table 8-1 Timer Configurations and Functions

Timer name	8/16 bits	Auto-reload	External event input	Timer output	PWM clock output	Baud rate generator	Watchdog timer
Timer 0	16	√	√	√	—	—	—
Timer 1	8/16	√	√	√	—	—	—
Timer 2	8/16	√	√	√	—	—	—
Timer 3	8	√	—	—	—	√ (SIO6)	—
Timer 4	8	√	—	√	—	√ (SIO1)	—
Timer 5	8	√	—	—	—	√ (SIO4, 5)	—
Timer 6	8	√	—	—	—	—	√
Timer 9	8	√	—	—	√	—	—

[Note]
TMOUT of timers 3, 5 and 9 are not output on ports, but can be used by software as a flag.

8.3 General-purpose 8-bit/16-bit Timer Registers

Table 8-2 lists a summary of SFRs for the control of general-purpose 8-bit and 16-bit timers.

Table 8-2 Summary of SFRs for General-Purpose 8-bit/16-bit Timer Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0062 0063	General-purpose 16-bit timer 0 counter	—	TM0C	R/W	16	Undefined	8-4
0064 0065	General-purpose 16-bit timer 0 register	—	TM0R	R/W	16	Undefined	8-4
0066☆	General-purpose 16-bit timer 0 control register	TM0CON	—	R/W	16	70	8-4
0068 0069	General-purpose 8-bit timer 12 counter	TM1C TM2C	TM12C	R/W	8/16	Undefined	8-10
006A 006B	General-purpose 8-bit timer control register	TM1R TM2R	TM12R	R/W	8/16	Undefined	8-10
006C☆	General-purpose 8-bit timer 1 control register	TM1CON	—	R/W	8	70	8-10
006D☆	General-purpose 8-bit timer 2 control register	TM2CON	—	R/W	8	40	8-11
0070	General-purpose 8-bit timer 3 counter	TM3C	—	R/W	8	Undefined	8-22
0071	General-purpose 8-bit timer 3 register	TM3R	—	R/W	8	Undefined	8-22
0072☆	General-purpose 8-bit timer 3 control register	TM3CON	—	R/W	8	70	8-22
0074	General-purpose 8-bit timer 4 counter	TM4C	—	R/W	8	Undefined	8-28
0075	General-purpose 8-bit timer 4 register	TM4R	—	R/W	8	Undefined	8-28
0076☆	General-purpose 8-bit timer 4 control register	TM4CON	—	R/W	8	70	8-28
0078	General-purpose 8-bit timer 5 counter	TM5C	—	R/W	8	Undefined	8-34
0079	General-purpose 8-bit timer 5 register	TM5R	—	R/W	8	Undefined	8-34
007A☆	General-purpose 8-bit timer 5 control register	TM5CON	—	R/W	8	Undefined	8-34
007C	General-purpose 8-bit timer 6 counter	TM6C	—	R/W	8	Undefined	8-40
007D	General-purpose 8-bit timer 6 register	TM6R	—	R/W	8	Undefined	8-40
007E☆	General-purpose 8-bit timer 6 control register	TM6CON	—	R/W	8	10	8-41
00CC	General-purpose 8-bit timer 9 counter	TM9C	—	R/W	8	Undefined	8-49
00CD	General-purpose 8-bit timer 9 register	TM9R	—	R/W	8	Undefined	8-49
00CE☆	General-purpose 8-bit timer 9 control register	TM9CON	—	R/W	8	70	8-49

[Notes]

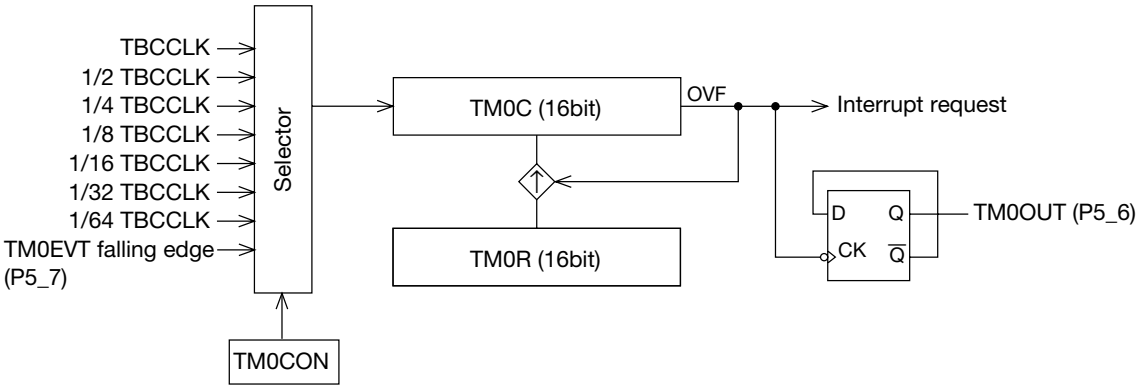
1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".
4. Bits 5 and 6 of the TM6COM register allow read only access (W is invalid). Bits 0 to 3 and 7 allow R/W access.

8.4 Timer 0

Timer 0 is a 16-bit auto-reload timer that has functions for external event input and timer output.

8.4.1 Timer 0 Configuration

Figure 8-1 shows the timer 0 configuration.



TM0C: General-purpose 16-bit timer 0 counter
TM0R: General-purpose 16-bit timer 0 register
TM0CON: General-purpose 16-bit timer 0 control register
TM0EVT: Timer 0 external event input pin (P5_7)
TM0OUT: Timer 0 output pin (P5_6)

Figure 8-1 Timer 0 Configuration

8.4.2 Description of Timer 0 Registers

(1) General-purpose 16-bit timer 0 counter (TM0C)

The general-purpose 16-bit timer 0 counter (TM0C) is a 16-bit up-counter. When this counter overflows, an interrupt request is generated and it is loaded with the contents of general-purpose 16-bit timer 0 register (TM0R).

TM0C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM0C are undefined.

[Note]
Writing a timer value to TM0C causes the same value to also be written to the general-purpose 16-bit timer 0 register (TM0R).

(2) General-purpose 16-bit timer 0 register (TM0R)

The general-purpose 16-bit timer 0 register (TM0R) consists of 16 bits. This register stores the value to be reloaded into the general-purpose 16-bit timer 0 counter (TM0C).

TM0R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM0R are undefined.

(3) General-purpose 16-bit timer 0 control register (TM0CON)

The general-purpose 16-bit timer 0 control register (TM0CON) consists of 5 bits. Bits 0 to 2 (TM0C0 to TM0C2) of TM0CON select the timer 0 count clock, bit 3 (TM0RUN) starts or halts the counting, and bit 7 (TM0OUT) specifies the initial timer output level (High or Low) at start-up.And each time TM0C overflows, the content of bit 7 (TM0OUT) is reversed.

TM0CON can be read from and written to by the program. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bits 4 to 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM0CON becomes 70H.
Figure 8-2 shows the TM0CON configuration.

[Note]
Just before TM0C overflows, if an SB, RB, XORB or other read-modify-write instruction is performed on TM0CON, then TM0OUT may not operate correctly.

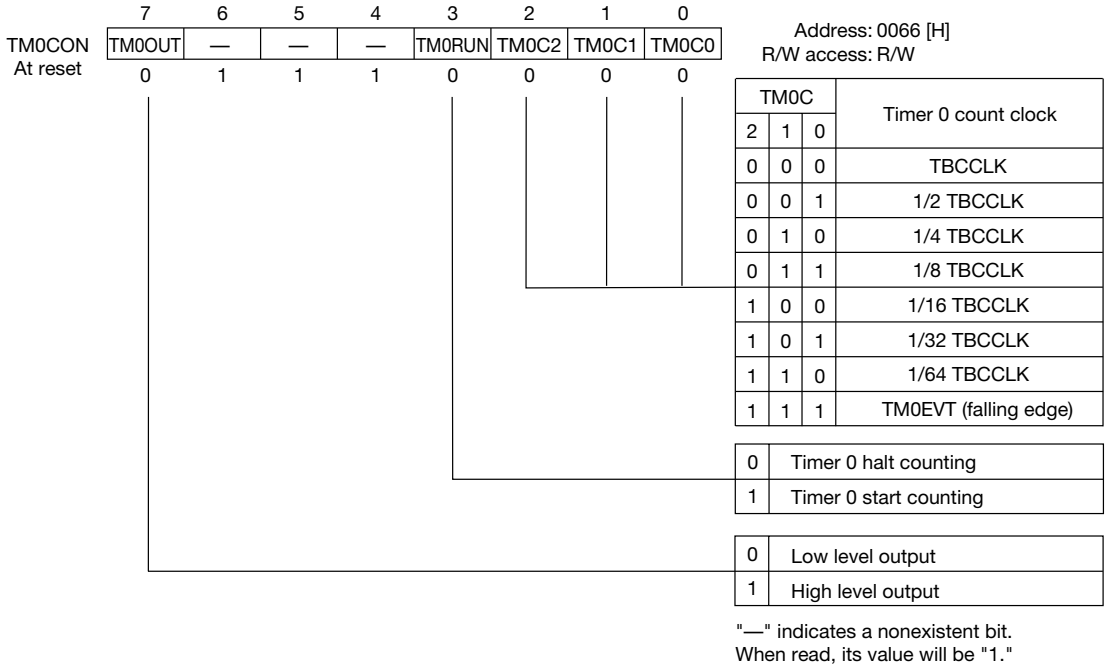


Figure 8-2 TM0CON Configuration

8.4.3 Example of Timer 0-related Register Settings

(1) Port 5 mode register (P5IO)

If TM0OUT (timer output) is to be used, set bit 6 (P5IO6) to "1" to configure the port as an output. If TM0EVT (event input) is to be used, reset bit 7 (P5IO7) to "0" to configure the port as an input.

(2) Port 5 secondary function control register (P5SF)

If TM0OUT (timer output) is to be used, set bit 6 (P5SF6) to "1" to configure the port as a secondary function output. If TM0EVT (event input) is to be used, disable or enable the pull-up resistor with bit 7 (P5SF7).

(3) General-purpose 16-bit timer 0 counter (TM0C)

Set the timer value that will be valid at the start of counting. When writing to TM0C, the same value will also be simultaneously and automatically written to the general-purpose 16-bit timer 0 register (TM0R).

(4) General-purpose 16-bit timer 0 register (TM0R)

This register sets the value to be loaded after general-purpose 16-bit timer 0 counter (TM0C) overflows. If the timer value (TM0C) and the reload value (TM0R) are identical, this register will automatically be set just by setting TM0C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 16-bit timer 0 control register (TM0CON)

Bits 0 to 2 (TM0C0 to TM0C2) of this register set the count clock for timer 0. If TM0OUT (timer output) is to be used, specify the initial value with bit 7 (TM0OUT). If bit 3 (TM0RUN) is set to "1", timer 0 will begin counting. If reset to "0", timer 0 will halt counting.

8.4.4 Timer 0 Operation

When the TM0RUN bit is set to "1", timer 0 will begin counting upward, running on the count clock selected by TM0CON. If external event input is selected as the count clock, timer 0 can also be used as an event counter. When TM0C overflows, an interrupt request is generated, the contents of TM0R are loaded into TM0C and the TM0OUT output is inverted. The initial value of the TM0OUT pin is specified by bit 7 (TM0OUT) of TM0CON. This operation is repeated until the TM0RUN bit is reset to "0". Figure 8-3 shows an operation example (for settings of 1/n counter frequency division ratio 1/1and 1/4 TBCCLK).

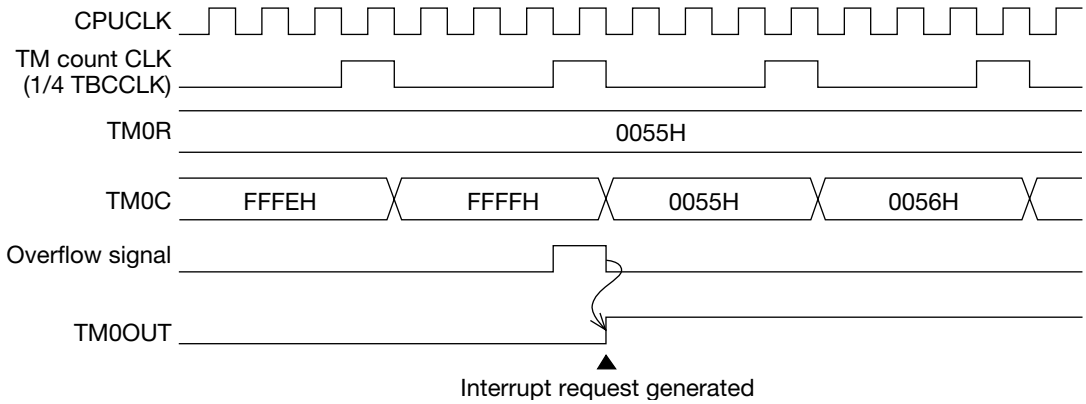


Figure 8-3 Timer 0 Operation

[Note]

Set the minimum pulse width of the external event input to at least 1 CPU clock (CPUCLK). The external event input signal is sampled at the falling edge of the CPUCLK to create the count clock for the timer.

8.4.5 Timer 0 Interrupt

When a timer 0 interrupt factor occurs, the interrupt request flag (QTM0OV) is set to "1". The interrupt request flag (QTM0OV) is located in interrupt request register 1 (IRQ1).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM0OV). The interrupt enable flag (ETM0OV) is located in interrupt enable register 1 (IE1).

Three levels of priority can be set with the interrupt priority setting flags (P0TM0OV and P1TM0OV). The interrupt priority setting flags are located in interrupt priority control register 2 (IP2).

Table 8-3 lists the vector address of the timer 0 interrupt factor and the interrupt processing flags.

Table 8-3 Timer 0 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 0	001A	QTM0OV	ETM0OV	P1TM0OV	P0TM0OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ1	IE1	IP2	
	Reference page	17-13	17-18	17-24	

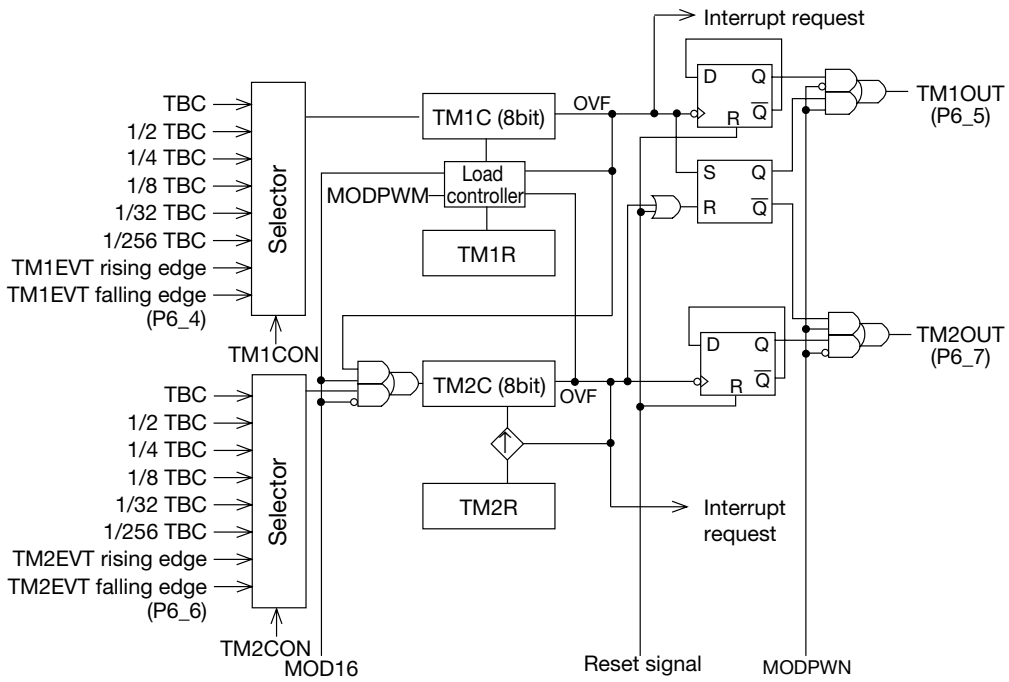
For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.5 Timers 1 and 2

Timers 1 and 2 are 8-bit auto-reload timers. Timers 1 and 2 can be combined and used in a 16-bit auto-reload mode. Timers 1 and 2 have functions for external event input, timer output, and PWM mode.

8.5.1 Timers 1 and 2 Configurations

Figure 8-4 shows the configuration of timers 1 and 2.



TM1C: General-purpose 8-bit timer 1 counter
TM2C: General-purpose 8-bit timer 2 counter
TM1R: General-purpose 8-bit timer 1 register
TM2R: General-purpose 8-bit timer 2 register
TM1CON: General-purpose 8-bit timer 1 control register
TM2CON: General-purpose 8-bit timer 2 control register
TM1EVT: Timer 1 external event input pin (P6_4)
TM2EVT: Timer 2 external event input pin (P6_6)
TM1OUT: Timer 1 output pin (P6_5)
TM2OUT: Timer 2 output pin (P6_7)

Figure 8-4 Timer 1 and 2 Configurations

8.5.2 Description of Timer 1 and 2 Registers

(1) General-purpose 8-bit timer 1 and 2 counters (TM1C, TM2C)

The general-purpose 8-bit timer 1 and 2 counters (TM1C, TM2C) are 8-bit up-counters. When each counter overflows, an interrupt request is generated and that counter is loaded with the contents of the general-purpose 8-bit timer 1 or 2 register (TM1R, TM2R).

TM1C and TM2C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM1C and TM2C are undefined.

[Note]
Writing a timer value to TM1C or TM2C causes the same value to also be written to the general-purpose 8-bit timer 1 and 2 registers (TM1R, TM2R).

(2) General-purpose 8-bit timer 1 and 2 registers (TM1R, TM2R)

The general-purpose 8-bit timer 1 and 2 registers (TM1R, TM2R) consist of 8 bits. These registers store the value to be reloaded into the general-purpose 8-bit timer 1 or 2 counter (TM1C, TM2C).

TM1R and TM2R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM1R and TM2R are undefined.

(3) General-purpose 8-bit timer 1 control register (TM1CON)

The general-purpose 8-bit timer 1 control register (TM1CON) consists of 5 bits. Bits 0 to 2 (TM1C0 to TM1C2) of TM1CON select the timer 1 count clock, bit 3 (TM1RUN) starts or stops the counting, and bit 7 (TM1OUT) specifies the initial timer output level (High or Low) at start-up. The value of bit 7 (TM1OUT) is inverted when TM1C overflows.

TM1CON can be read from and written to by the program. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bits 4 to 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM1CON becomes 70H.

Figure 8-5 shows the TM1CON configuration.

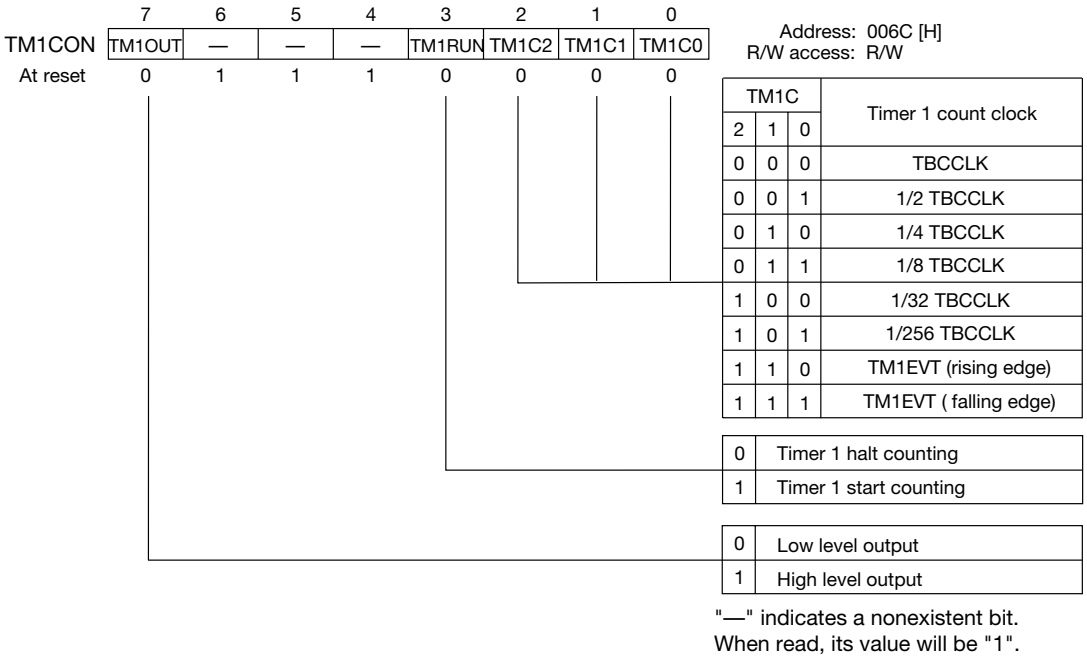


Figure 8-5 TM1CON Configuration

(4) General-purpose 8-bit timer 2 control register (TM2CON)

The general-purpose 8-bit timer 2 control register (TM2CON) consists of 7 bits. TM2CON can be read from and written to by the program. However, write operation are invalid for bit 6. If read, a value of "1" will always be obtained for bit 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM2CON becomes 40H.

Figure 8-6 shows the TM2CON configuration.

[Description of each bit]

- TM2C0 to TM2C2 (bits 0 to 2)
  These bits specify the timer 2 count clock.
- TM2RUN (bit 3)
  This bit starts or stops the counting.
- MOD16 (bit 4)
  Setting this bit combines timer 1 and 2 into the 16-bit auto-reload mode. While this bit is set, the settings of TM2C0 to TM2C2 and TM2RUN are invalid.
- MODPWM (bit 5)
  Setting this bit combines timer 1 and 2 into the PWM mode. While this bit is set, the setting of TM2RUN is invalid, and if TM1RUN is set, timer 1 and 2 will count simultaneously.
- TM2OUT (bit 7)
  This bit specifies the initial timer output level (High or Low) at start-up. The value of bit 7 (TM2OUT) is inverted whenever TM2C overflows.

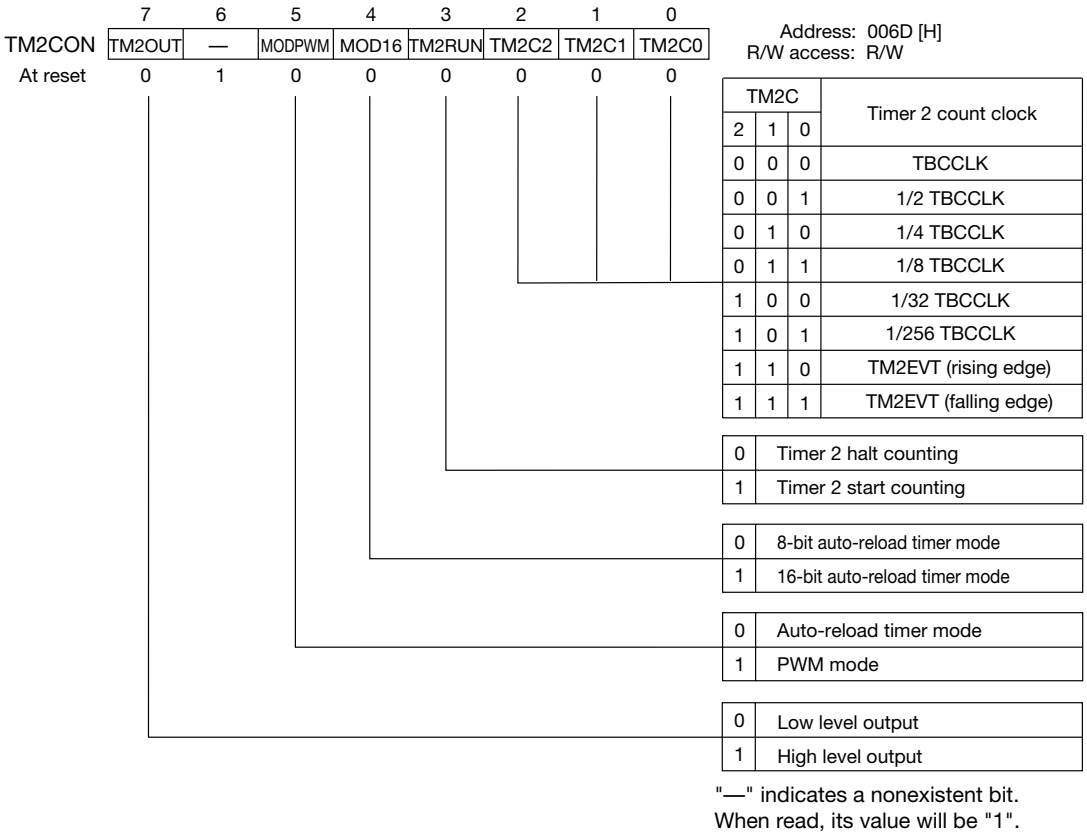


Figure 8-6 TM2CON Configuration

8.5.3 Example of Timer 1- and 2-related Register Settings

- **8-bit auto-reload timer mode (Timer 1)**

(1) Port 6 mode register (P6IO)

If TM1OUT (timer output) is to be used, set bit 5 (P6IO5) to "1" to configure the port as an output. If TM1EVT (event input) is to be used, reset bit 4 (P6IO4) to "0" to configure the port as an input.

(2) Port 6 secondary function control register (P6SF)

If TM1OUT (timer output) is to be used, set bit 5 (P6SF5) to "1" to configure the port as a secondary function output. If TM1EVT (event input) is to be used, disable or enable the pull-up resistor with bit 4 (P6SF4).

(3) General-purpose 8-bit timer 1 counter (TM1C)

Set the timer value that will be valid at the start of counting. When writing to TM1C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 1 register (TM1R).

(4) General-purpose 8-bit timer 1 register (TM1R)

This register sets the value to be loaded after general-purpose 8-bit timer 1 counter (TM1C) overflows. If the timer value (TM1C) and the reload value (TM1R) are identical, this register will automatically be set just by setting TM1C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 8-bit timer 1 control register (TM1CON)

Bits 9 to 0 (TM1C0 to TM1C2) of this register specify the count clock for timer 1. If TM1OUT (timer output) is to be used, specify the initial value with bit 7 (TM1OUT). If bit 3 (TM1RUN) is set to "1", timer 1 will begin counting. If reset to "0", timer 1 will halt counting.

- **8-bit auto-reload timer mode (Timer 2)**

(1) Port 6 mode register (P6IO)

If TM2OUT (timer output) is to be used, set bit 7 (P6IO7) to "1" to configure the port as an output. If TM2EVT (event input) is to be used, reset bit 6 (P6IO6) to "0" to configure the port as an input.

(2) Port 6 secondary function control register (P6SF)

If TM2OUT (timer output) is to be used, set bit 7 (P6SF7) to "1" to configure the port as a secondary function output. If TM2EVT (event input) is to be used, disable or enable the pull-up resistor with bit 6 (P6SF6).

(3) General-purpose 8-bit timer 2 counter (TM2C)

Set the timer value that will be valid at the start of counting. When writing to TM2C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 2 register (TM2R).

(4) General-purpose 8-bit timer 2 register (TM2R)

This register sets the value to be loaded after general-purpose 8-bit timer 2 counter (TM2C) overflows. If the timer value (TM2C) and the reload value (TM2R) are identical, this register will automatically be set just by setting TM2C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 8-bit timer 2 control register (TM2CON)

Bits 0 to 2 (TM2C0 to TM2C2) of this register specify the count clock for timer 2. If TM2OUT (timer output) is to be used, specify the initial value with bit 7 (TM2OUT). If bit 3 (TM2RUN) is set to "1", timer 2 will begin counting. If reset to "0", timer 2 will halt counting.

- **16-bit auto-reload timer mode**

(1) Port 6 mode register (P6IO)

If TM1OUT (timer 1 output) and TM2OUT (timer 2 output) are to be used, set bits 5 and 7 (P6IO5, P6IO7) to "1" to configure the ports as outputs. If TM1EVT (event input) is to be used, reset bit 4 (P6IO4) to "0" to configure the port as an input.

(2) Port 6 secondary function control register (P6SF)

If TM1OUT (timer 1 output) and TM2OUT (timer 2 output) are to be used, set bits 5 and 7 (P6SF5, P6SF7) to "1" to configure the ports as secondary function outputs. If TM1EVT (event input) is to be used, disable or enable the pull-up resistor with bit 4 (P6SF4).

(3) General-purpose 16-bit timer 12 counter (TM12C)

Set the timer value that will be valid at the start of counting. When writing to TM12C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 12 register (TM12R).

(4) General-purpose 16-bit timer 12 register (TM12R)

This register sets the value to be loaded after general-purpose 16-bit timer 12 counter (TM12C) overflows. If the timer value (TM12C) and the reload value (TM12R) are identical, this register will automatically be set just by setting TM12C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 8-bit timer 1 control register (TM1CON)

Bits 0 to 2 (TM1C0 to TM1C2) of this register specify the count clock for timer 1. If TM1OUT (timer 1 output) is to be used, specify the initial value with bit 7 (TM1OUT). If bit 3 (TM1RUN) is set to "1", timer 1 will begin counting. If reset to "0", timer 1 will halt counting.

(6) General-purpose 8-bit timer 2 control register (TM2CON)

Setting bit 4 (MOD16) to "1" sets the 16-bit timer mode. While this bit is set, bits 0 to 2 (TM2C0 to TM2C2) and bit 3 (TM2RUN) settings are invalid and setting bit 3 (TM1RUN) of the timer 1 control register (TM1CON) to "1" starts simultaneous counting of timers 1 and 2. If TM2OUT (timer 2 output) is to be used, specify the initial value with bit 7 (TM2OUT).

- **PWM mode**

(1) Port 6 mode register (P6IO)

If TM1OUT (timer 1 output) and TM2OUT (timer 2 output) are to be used, set bits 5 and 7 (P6IO5, P6IO7) to "1" to configure the ports as outputs. If TM1EVT and TM2EVT (event inputs) are to be used, reset bit 4 and 6 (P6IO4, P6IO6) to "0" to configure the ports inputs.

(2) Port 6 secondary function control register (P6SF)

If TM1OUT (timer 1 output) and TM2OUT (timer 2 output) are to be used, set bits 5 and 7 (P6SF5, P6SF7) to "1" to configure the ports as secondary function outputs. If TM1EVT and TM2EVT (event input) are to be used, disable or enable the pull-up resistor with bits 4 and 6 (P6SF4, P6SF6).

(3) General-purpose 16-bit timer 12 counter (TM12C)

Set the timer value that will be valid at the start of counting. When writing to TM12C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 12 register (TM12R).

(4) General-purpose 16-bit timer 12 register (TM12R)

This register sets the value to be loaded after general-purpose 16-bit timer 12 counter (TM12C) overflows. If the timer value (TM12C) and the reload value (TM12R) are identical, this register will automatically be set just by setting TM12C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 8-bit timer 1 control register (TM1CON)

Bits 0 to 2 (TM1C0 to TM1C2) of this register specify the count clock for timer 1. If TM1OUT (timer 1 output) is to be used, specify the initial value with bit 7 (TM1OUT). If bit 3 (TM1RUN) is set to "1", timer 1 will begin counting. If reset to "0", timer 1 will halt counting.

(6) General-purpose 8-bit timer 2 control register (TM2CON)

Setting bit 5 (MODPWM) to "1" sets the PWM mode. While this bit is set, bit 3 (TM2RUN) settings are invalid; setting bit 3 (TM1RUN) of the timer 1 control register (TM1CON) to "1", starts simultaneous counting of timers 1 and 2. If TM2OUT (timer 2 output) is to be used, specify the initial value with bit 7 (TM2OUT).

8.5.4 Timer 1 and 2 Operation

- **8-bit auto-reload timer mode**

  When the RUN bits corresponding to TM1 and TM2 are set to "1", timers 1 and 2 will begin counting upward, running on the count clocks selected by TM1CON and TM2CON. When TM1C and TM2C overflow, individual interrupt requests are generated, and the corresponding contents of TM1R and TM2R are loaded into TM1C and TM2C. In addition, the output of TM1OUT and TM2OUT is inverted. This operation is repeated until the RUN bits are reset to "0". Figure 8-7 shows an example of 8-bit auto-reload timer mode operation.

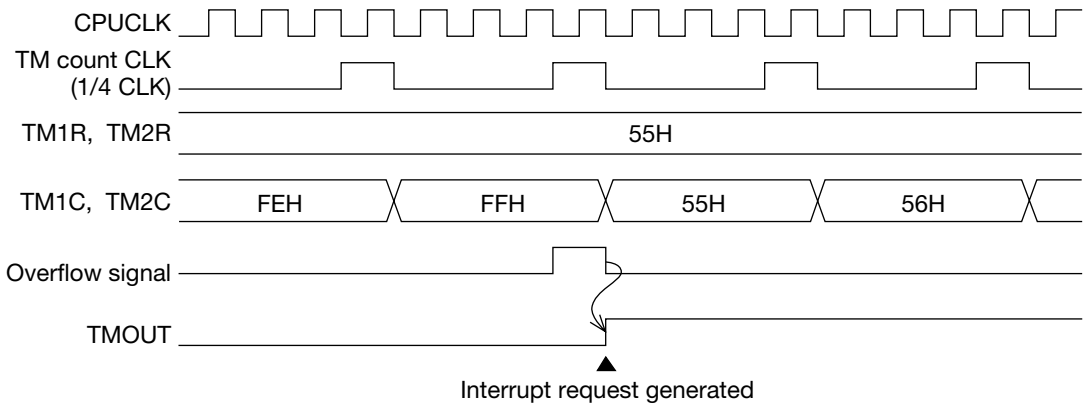


Figure 8-7 8-Bit Auto-Reload Timer Mode Operation Example

- **16-bit auto-reload timer mode**

  Setting the MOD16 bit of TM2CON to "1" combines TM1 and TM2 to set the 16-bit auto-reload timer mode. TM2C counts upward, using overflow of TM1C as the count clock. When TM2C overflows, a timer 2 interrupt request is generated, and the contents of TM1R and TM2R are loaded into TM1C and TM2C respectively. In addition, the output of TM2OUT is inverted.

  During this mode, overflow of TM1C does not cause the contents of TM1R to be loaded. However, a timer 1 interrupt request will be generated and the output of TM1OUT will change.

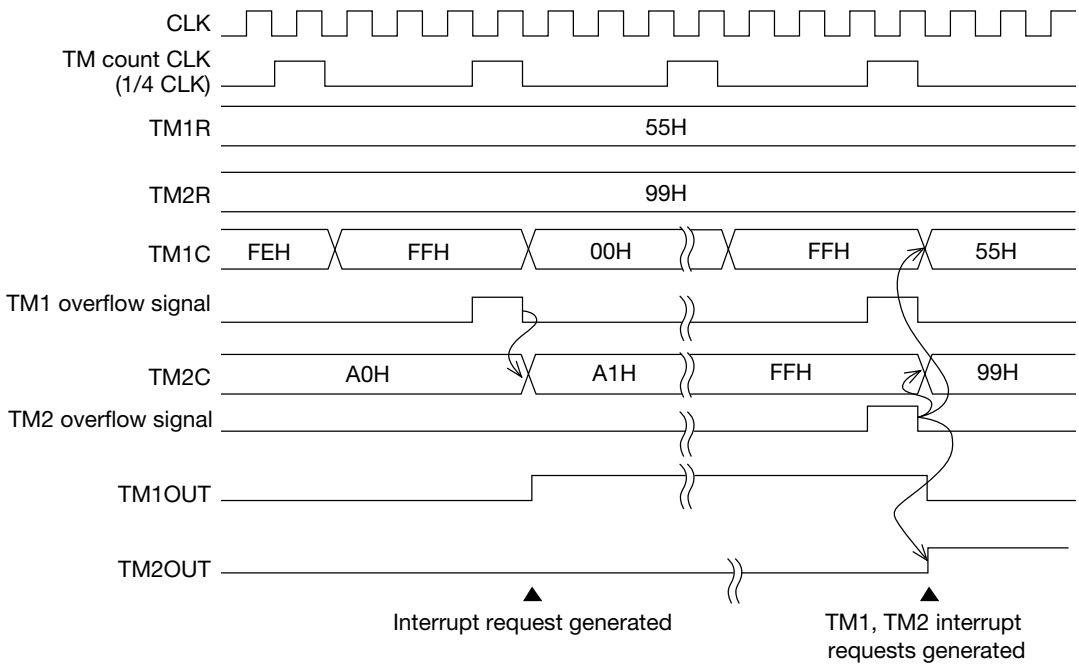


Figure 8-8 16-Bit Auto-Reload Timer Mode Operation Example

- **PWM mode**

  Setting the MODPWM bit of TM2CON to "1" sets the PWM mode that uses TM1 and TM2. During the PWM mode, since the following operation is performed, use TM2C as the PWM cycle counter and TM1C as the duty control counter.

  When TM1C overflows, an interrupt request is generated, and the PWM F/F is set. When TM2C overflows, an interrupt request is generated, the contents of TM1R and TM2R are loaded into TM1C and TM2C respectively, and the PWM F/F is reset. If "set" and "reset" of the PWM F/F are simultaneously generated, priority is given to the "reset".

  The Q output (positive phase) of the PWM F/F is output from TM1OUT and the Q output of the PWM F/F (inverted phase) is output from TM2OUT.

  Note that if the count clock selected for TM1C is faster than the TM2C count clock, interrupt requests due to TM1C overflow may occur two or more times in a single cycle.

  Also note that if the count clock selected for TM1C is slower than the TM2C count clock, at the start of counting (when TM1RUN is set), and when TM2C overflows, a synchronous shift will occur, and the TM1C overflow cycle may shift. (The same count clocks are recommended.)

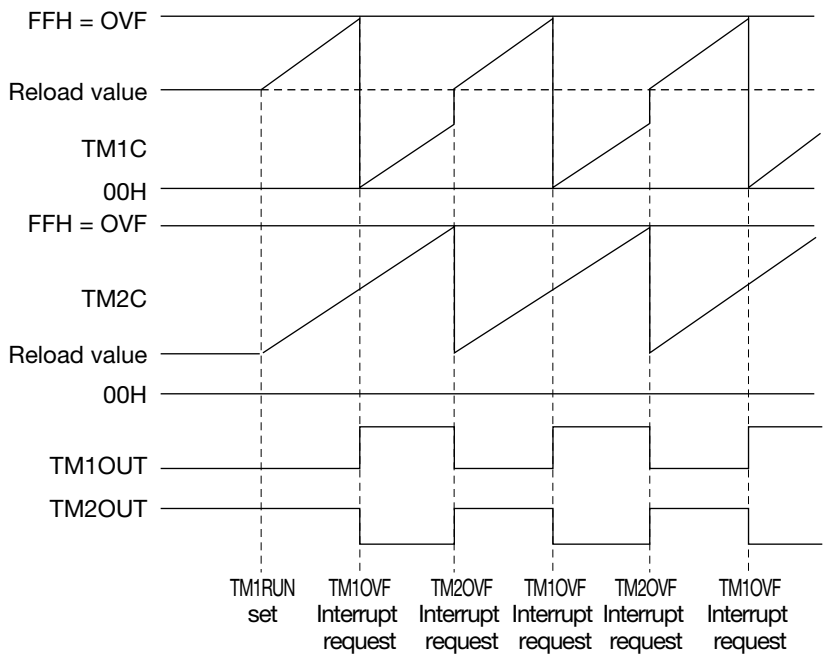


Figure 8-9 PWM Timer Mode Operation Example

8.5.5 Timer 1 and 2 Interrupts

- **Timer 1 interrupt**

  When a timer 1 interrupt factor occurs, the interrupt request flag (QTM1OV) is set to "1". The interrupt request flag (QTM1OV) is located in interrupt request register 1 (IRQ1).

  Interrupts can be enabled or disabled by the interrupt enable flag (ETM1OV). The interrupt enable flag (ETM1OV) is located in interrupt enable register 1 (IE1).

  Three levels of priority can be set with the interrupt priority setting flags (P0TM1OV and P1TM1OV). The interrupt priority setting flags are located in interrupt priority control register 3 (IP3).

  Table 8-4 lists the vector address and interrupt processing flags for the timer 1 interrupt factor.

Table 8-4 Timer 1 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 1	0022	QTM1OV	ETM1OV	P1TM1OV	P0TM1OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ1	IE1	IP3	
	Reference page	17-13	17-18	17-25	

For farther details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

- **Timer 2 interrupt**

  When a timer 2 interrupt factor occurs, the interrupt request flag (QTM2OV) is set to "1". The interrupt request flag (QTM2OV) is located in interrupt request register 1 (IRQ1).

  Interrupts can be enabled or disabled by the interrupt enable flag (ETM2OV). The interrupt enable flag (ETM2OV) is located in interrupt enable register 1 (IE1).

  Three levels of priority can be set with the interrupt priority setting flags (P0TM2OV and P1TM2OV). The interrupt priority setting flags are located in interrupt priority control register 3 (IP3).

  Table 8-5 lists the vector address and interrupt processing flags for the timer 2 interrupt factor.

Table 8-5 Timer 2 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 2	0024	QTM2OV	ETM2OV	P1TM2OV	P0TM2OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ1	IE1	IP3	
	Reference page	17-13	17-18	17-25	

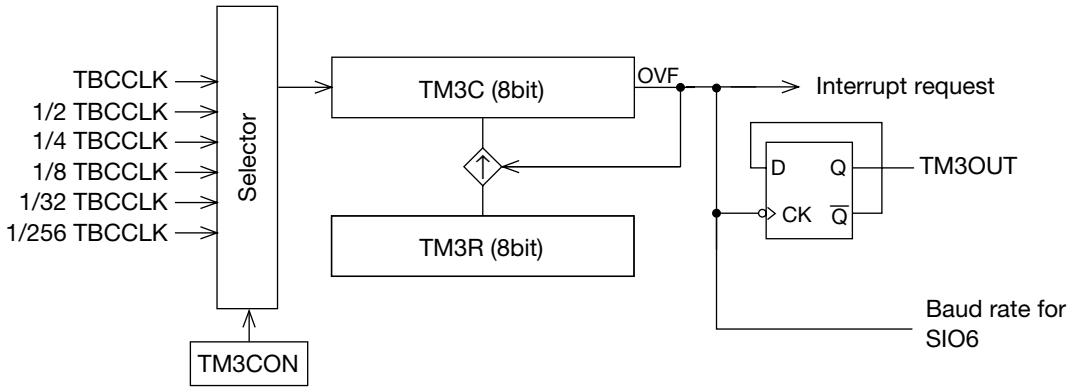
For farther details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.6 Timer 3

Timer 3 is an 8-bit auto-reload timer that has a baud rate generator function for SIO6. TM3OUT is not output on ports, but can be used by software as a flag.

8.6.1 Timer 3 Configuration

Figure 8-10 shows the timer 3 configuration.



TM3C: General-purpose 8-bit timer 3 counter
TM3R: General-purpose 8-bit timer 3 register
TM3CON: General-purpose 8-bit timer 3 control register

Figure 8-10 Timer 3 Configuration

8.6.2 Description of Timer 3 Registers

(1) General-purpose 8-bit timer 3 counter (TM3C)

The general-purpose 8-bit timer 3 counter (TM3C) is an 8-bit up-counter. When this counter overflows, an interrupt request is generated and it is loaded with the contents of general-purpose 8-bit timer 3 register (TM3R). TM3C can also be used as a baud rate generator for SIO6.

TM3C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM3C are undefined.

[Note]
Writing a timer value to TM3C causes the same value to also be written to the general-purpose 8-bit timer 3 register (TM3R).

(2) General-purpose 8-bit timer 3 register (TM3R)

The general-purpose 8-bit timer 3 register (TM3R) consists of 8 bits. This register stores the value to be reloaded into the general-purpose 8-bit timer 3 counter (TM3C).

TM3R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM3R are undefined.

(3) General-purpose 8-bit timer 3 control register (TM3CON)

The general-purpose 8-bit timer 3 control register (TM3CON) consists of 5 bits. Bits 0 to 2 (TM3C0 to TM3C2) of TM3CON select the timer 3 count clock, bit 3 (TM3RUN) starts or halts the counting, and bit 7 (TM3OUT) specifies the initial timer output level (High or Low) at start-up. And each time TM3C overflows, the content of bit 7 (TM3OUT) is reversed.

TM3CON can be read from and written to by the program. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bits 4 to 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM3CON becomes 70H.

Figure 8-11 shows the TM3CON configuration.

[Note]
Just before TM3C overflows, if an SB, RB, XORB or other read-modify-write instruction is performed on TM3CON, then TM3OUT may not operate correctly.

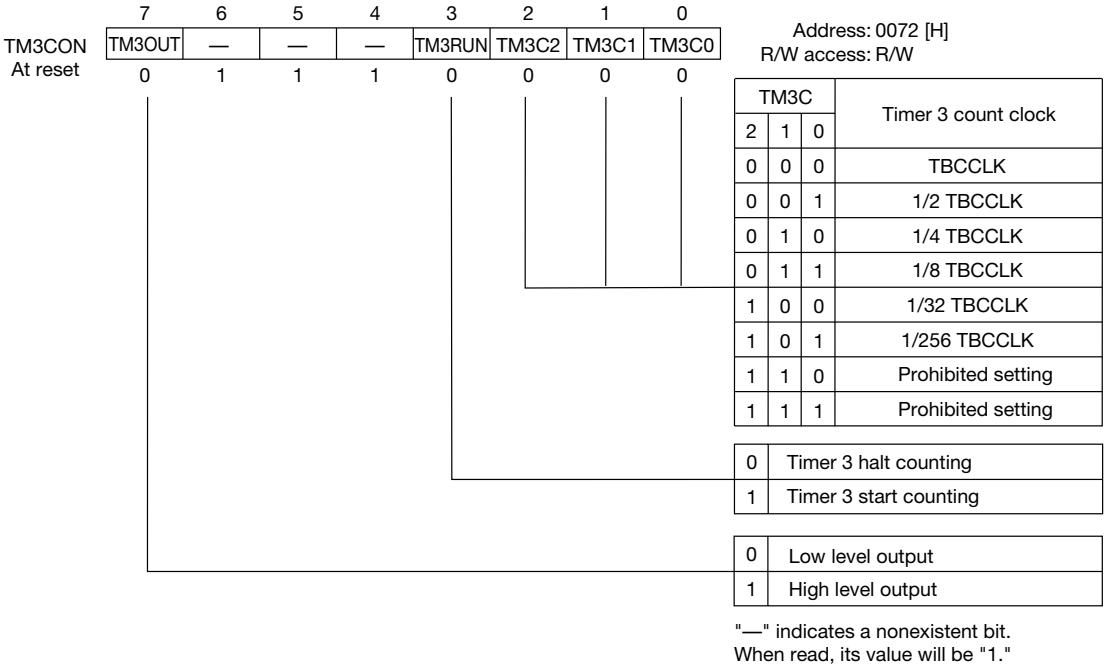


Figure 8-11 TM3CON Configuration

[Note]

Do not select a timer 3 count clock setting that is prohibited. If a "prohibited setting" is selected, timer 3 will not operate properly.

8.6.3 Example of Timer 3-related Register Settings

(1) General-purpose 8-bit timer 3 counter (TM3C)

Set the timer value that will be valid at the start of counting. When writing to TM3C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 3 register (TM3R).

(2) General-purpose 8-bit timer 3 register (TM3R)

This register sets the value to be loaded after general-purpose 8-bit timer 3 counter (TM3C) overflows. If the timer value (TM3C) and the reload value (TM3R) are identical, this register will automatically be set just by setting TM3C. If the values are different or are to be modified, this register must be set explicitly.

(3) General-purpose 8-bit timer 3 control register (TM3CON)

Bits 0 to 2 (TM3C0 to TM3C2) of this register specify the count clock for timer 3. If TM3OUT (timer output) is to be used, specify the initial value with bit 7 (TM3OUT). If bit 3 (TM3RUN) is set to "1", timer 3 will begin counting. If reset to "0", timer 3 will halt counting.

8.6.4 Timer 3 Operation

When the TM3RUN bit is set to "1", timer 3 will begin counting upward, running on the count clock selected by TM3CON. When TM3C overflows, an interrupt request is generated, the contents of TM3R are loaded into TM3C and the TM3OUT output is inverted. The initial value of the TM3OUT pin is specified by bit 7 (TM3OUT) of TM3CON. This operation is repeated until the TM3RUN bit is reset to "0". Overflow of TM3C can be used as a baud rate generator for SIO6. Figure 8-12 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

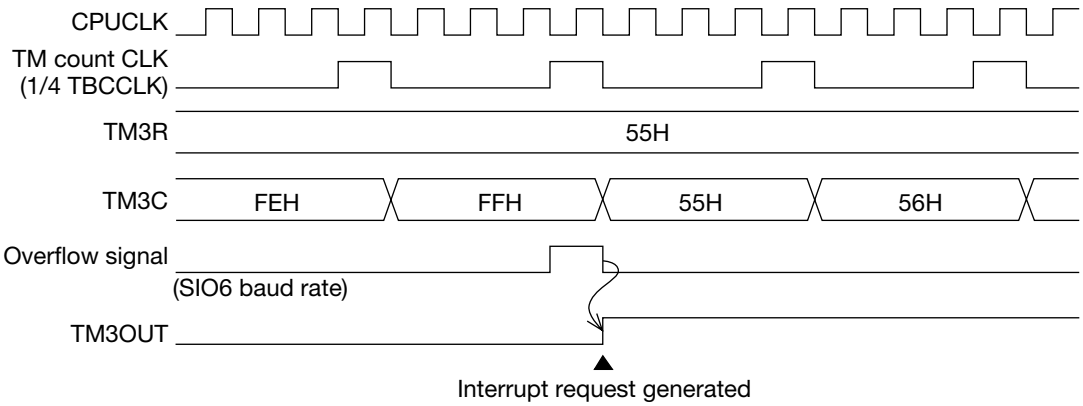


Figure 8-12 Timer 3 Operation Example

8.6.5 Timer 3 Interrupt

When a timer 3 interrupt factor occurs, the interrupt request flag (QTM3OV) is set to "1". The interrupt request flag (QTM3OV) is located in interrupt request register 1 (IRQ1).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM0OV). The interrupt enable flag (ETM3OV) is located in interrupt enable register 1 (IE1).

Three levels of priority can be set with the interrupt priority setting flags (P0TM3OV and P1TM3OV). The interrupt priority setting flags are located in interrupt priority control register 3 (IP3).

Table 8-6 lists the vector address of the timer 3 interrupt factor and the interrupt processing flags.

Table 8-6 Timer 3 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 3	0026	QTM3OV	ETM3OV	P1TM3OV	P0TM3OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ1	IE1	IP3	
	Reference page	17-13	17-18	17-25	

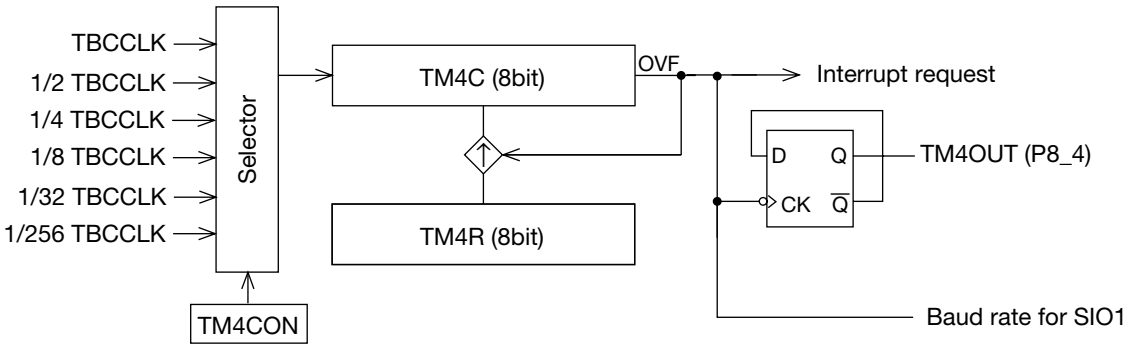
For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.7 Timer 4

Timer 4 is an 8-bit auto-reload timer that has functions for timer output and a baud rate generator for SIO1.

8.7.1 Timer 4 Configuration

Figure 8-13 shows the timer 4 configuration.



TM4C: General-purpose 8-bit timer 4 counter
TM4R: General-purpose 8-bit timer 4 register
TM4CON: General-purpose 8-bit timer 4 control register
TM4OUT: Timer 4 output pin (P8_4)

Figure 8-13 Timer 4 Configuration

8.7.2 Description of Timer 4 Registers

(1) General-purpose 8-bit timer 4 counter (TM4C)

The general-purpose 8-bit timer 4 counter (TM4C) is an 8-bit up-counter. When this counter overflows, an interrupt request is generated and it is loaded with the contents of general-purpose 8-bit timer 4 register (TM4R). TM4C can also be used as a baud rate generator for SIO1.

TM4C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM4C are undefined.

[Note]
Writing a timer value to TM4C causes the same value to also be written to the general-purpose 8-bit timer 4 register (TM4R).

(2) General-purpose 8-bit timer 4 register (TM4R)

The general-purpose 8-bit timer 4 register (TM4R) consists of 8 bits. This register stores the value to be reloaded into the general-purpose 8-bit timer 4 counter (TM4C).

TM4R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM4R are undefined.

(3) General-purpose 8-bit timer 4 control register (TM4CON)

The general-purpose 8-bit timer 4 control register (TM4CON) consists of 5 bits. Bits 0 to 2 (TM4C0 to TM4C2) of TM4CON select the timer 4 count clock, bit 3 (TM4RUN) starts or halts the counting, and bit 7 (TM4OUT) specifies the initial timer output level (High or Low) at start-up. And each time TM4C overflows, the content of bit 7 (TM4OUT) is reversed.

TM4CON can be read from and written to by the program. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bits 4 to 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM4CON becomes 70H.

Figure 8-14 shows the TM4CON configuration.

[Note]
Just before TM4C overflows, if an SB, RB, XORB or other read-modify-write instruction is performed on TM4CON, then TM4OUT may not operate correctly.

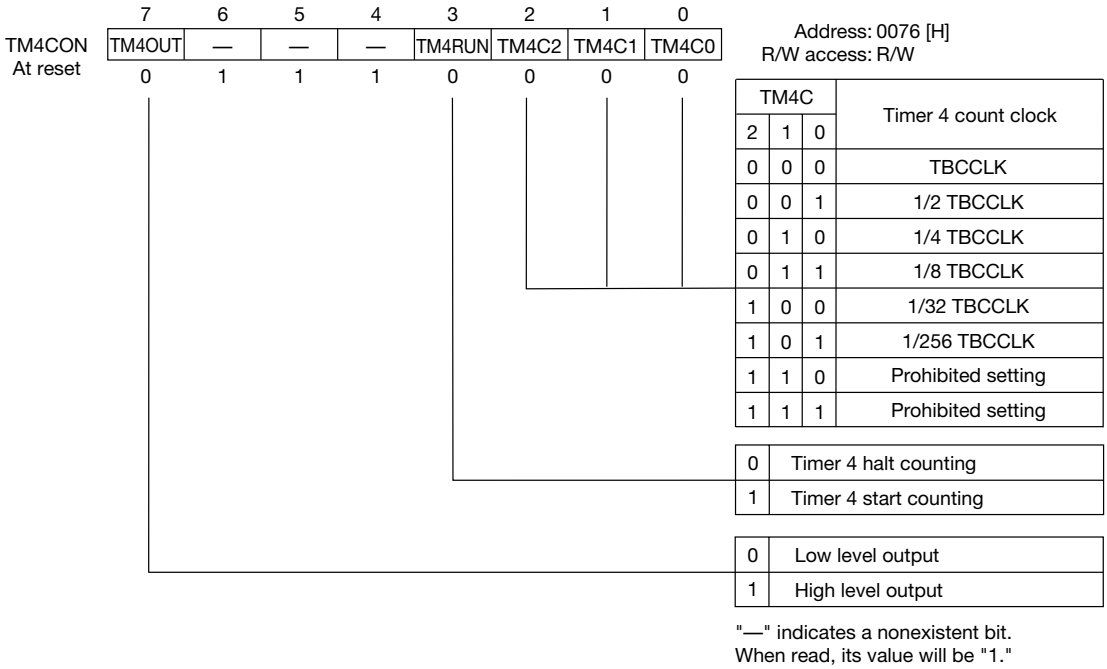


Figure 8-14 TM4CON Configuration

[Note]

Do not select a timer 4 count clock setting that is prohibited. If a "prohibited setting" is selected, timer 4 will not operate properly.

8.7.3 Example of Timer 4-related Register Settings

(1) Port 8 mode register (P8IO)

If TM4OUT (timer output) is to be used, set bit 4 (P8IO4) to "1" to configure the port as an output.

(2) Port 8 secondary function control register (P8SF)

If TM4OUT (timer output) is to be used, set bit 4 (P8SF4) to "1" to configure the port as a secondary function output.

(3) General-purpose 8-bit timer 4 counter (TM4C)

Set the timer value that will be valid at the start of counting. When writing to TM4C, the same value will also be written simultaneously and automatically to the general-purpose 8-bit timer 4 register (TM4R).

(4) General-purpose 8-bit timer 4 register (TM4R)

This register sets the value to be loaded after general-purpose 8-bit timer 4 counter (TM4C) overflows. If the timer value (TM4C) and the reload value (TM4R) are identical, this register will automatically be set just by setting TM4C. If the values are different or are to be modified, this register must be set explicitly.

(5) General-purpose 8-bit timer 4 control register (TM4CON)

Bits 0 to 2 (TM4C0 to TM4C2) of this register specify the count clock for timer 4. If TM4OUT (timer output) is to be used, specify the initial value with bit 7 (TM4OUT). If bit 3 (TM4RUN) is set to "1", timer 4 will begin counting. If reset to "0", timer 4 will halt counting.

8.7.4 Timer 4 Operation

When the TM4RUN bit is set to "1", timer 4 will begin counting upward, running on the count clock selected by TM4CON. When TM4C overflows, an interrupt request is generated, the contents of TM4R are loaded into TM4C and the TM4OUT output is inverted. The initial value of the TM4OUT pin is specified by bit 7 (TM4OUT) of TM4CON. This operation is repeated until the TM4RUN bit is reset to "0". Overflow of TM4C can be used as a baud rate generator for SIO1. Figure 8-15 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

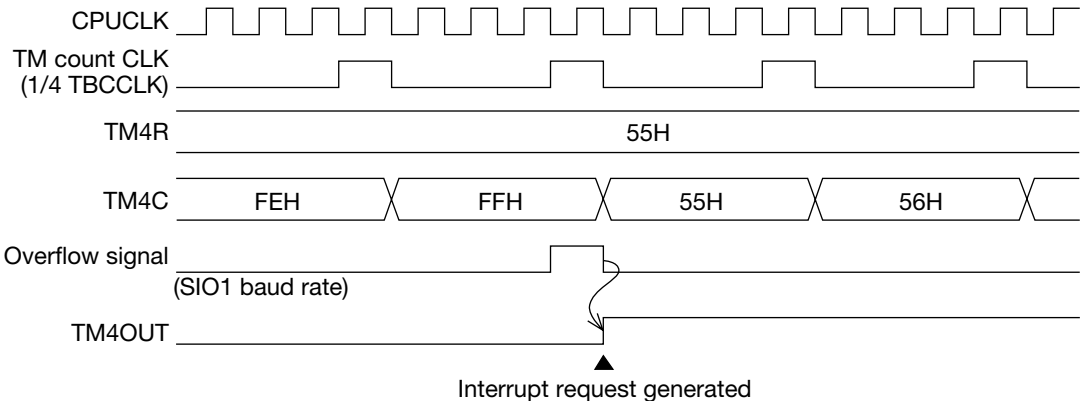


Figure 8-15 Timer 4 Operation Example

8.7.5 Timer 4 Interrupt

When a timer 4 interrupt factor occurs, the interrupt request flag (QTM4OV) is set to "1". The interrupt request flag (QTM4OV) is located in interrupt request register 2 (IRQ2).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM4OV). The interrupt enable flag (ETM4OV) is located in interrupt enable register 2 (IE2).

Three levels of priority can be set with the interrupt priority setting flags (P0TM4OV and P1TM4OV). The interrupt priority setting flags are located in interrupt priority control register 5 (IP5).

Table 8-7 lists the vector address of the timer 4 interrupt factor and the interrupt processing flags.

Table 8-7 Timer 4 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 4	0036	QTM4OV	ETM4OV	P1TM4OV	P0TM4OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ2	IE2	IP5	
	Reference page	17-14	17-19	17-27	

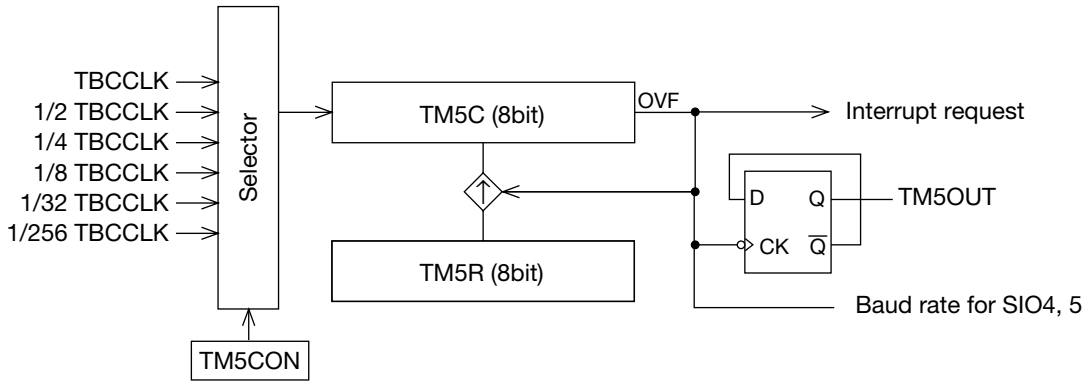
For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.8 Timer 5

Timer 5 is an 8-bit auto-reload timer that has a baud rate generator function for SIO4 and 5. TM5OUT is not output on ports, but can be used by software as a flag.

8.8.1 Timer 5 Configuration

Figure 8-16 shows the timer 5 configuration.



TM5C: General-purpose 8-bit timer 5 counter
TM5R: General-purpose 8-bit timer 5 register
TM5CON: General-purpose 8-bit timer 5 control register

Figure 8-16 Timer 5 Configuration

8.8.2 Description of Timer 5 Registers

(1) General-purpose 8-bit timer 5 counter (TM5C)

The general-purpose 8-bit timer 5 counter (TM5C) is an 8-bit up-counter. When this counter overflows, an interrupt request is generated and it is loaded with the contents of general-purpose 8-bit timer 5 register (TM5R). TM5C can also be used as a baud rate generator for SIO4 and 5.

TM5C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), contents of TM5C are undefined.

[Note]
Writing a timer value to TM5C causes the same value to also be written to the general-purpose 8-bit timer 5 register (TM5R).

(2) General-purpose 8-bit timer 5 register (TM5R)

The general-purpose 8-bit timer 5 register (TM5R) consists of 8 bits. This register stores the value to be reloaded into the general-purpose 8-bit timer 5 counter (TM5C).

TM5R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM5R are undefined.

(3) General-purpose 8-bit timer 5 control register (TM5CON)

The general-purpose 8-bit timer 5 control register (TM5CON) consists of 5 bits. Bits 0 to 2 (TM5C0 to TM5C2) of TM5CON select the timer 5 count clock and bit 3 (TM5RUN) starts or halts the counting. Specify the initial timer output level (High or Low) at start-up with bit 7 (TM5OUT). The contents of bit 7 (TM5OUT) are reversed each time TM5C overflows.

TM5CON can be read from and written to by the program. However, write operations are invalid for the upper 4 bits. If read, a value of "1" will always be obtained for bits 4 to 6. The value read from bit 7 is undefined.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM5CON are undefined.

Figure 8-17 shows the TM5CON configuration.

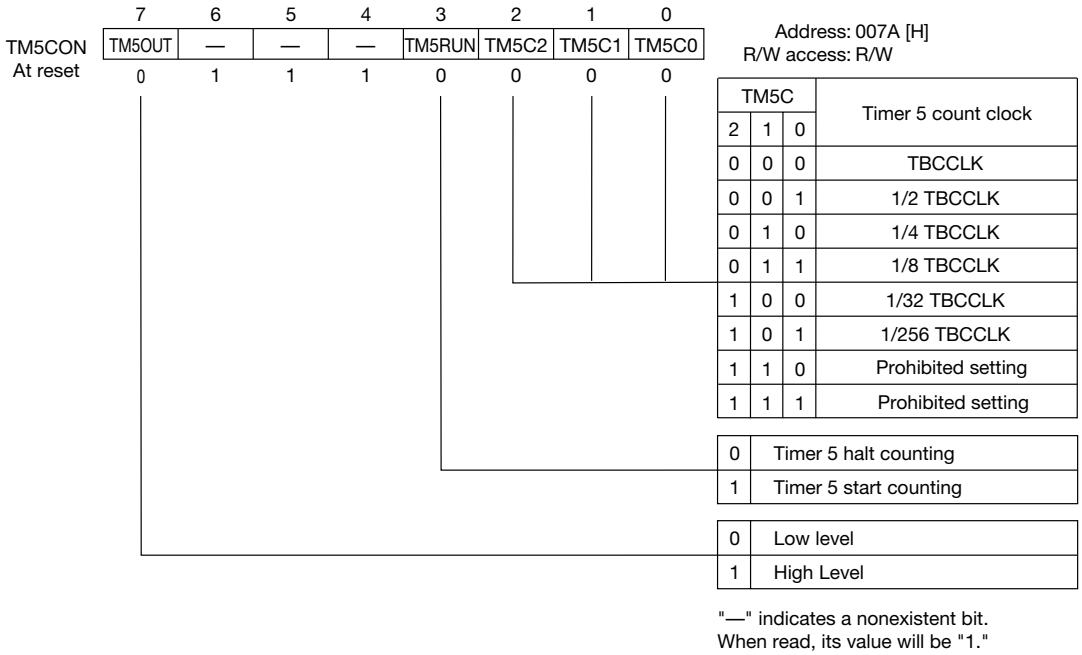


Figure 8-17 TM5CON Configuration

[Note]
Do not select a timer 5 count clock setting that is prohibited. If a "prohibited setting" is selected, timer 5 will not operate properly.

8.8.3 Example of Timer 5-related Register Settings

(1) General-purpose 8-bit timer 5 counter (TM5C)

Set the timer value that will be valid at the start of counting. When writing to TM5C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 5 register (TM5R).

(2) General-purpose 8-bit timer 5 register (TM5R)

This register sets the value to be loaded after general-purpose 8-bit timer 5 counter (TM5C) overflows. If the timer value (TM5C) and the reload value (TM5R) are identical, this register will automatically be set just by setting TM5C. If the values are different or are to be modified, this register must be set explicitly.

(3) General-purpose 8-bit timer 5 control register (TM5CON)

Bits 0 to 2 (TM5C0 to TM5C2) of this register specify the count clock for timer 5. If TM5OUT (timer output) is used, specify the initial value with bit 7 (TM5OUT). If bit 3 (TM5RUN) is set to "1", timer 5 will begin counting. If reset to "0", timer 5 will halt counting.

8.8.4 Timer 5 Operation

When the TM5RUN bit is set to "1", timer 5 will begin counting upward, running on the count clock selected by TM5CON. When TM5C overflows, an interrupt request is generated and the contents of TM5R are loaded into TM5C. This operation is repeated until the TM5RUN bit is reset to "0". Overflow of TM5C can be used as a baud rate generator for SIO4 and 5. Figure 8-18 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

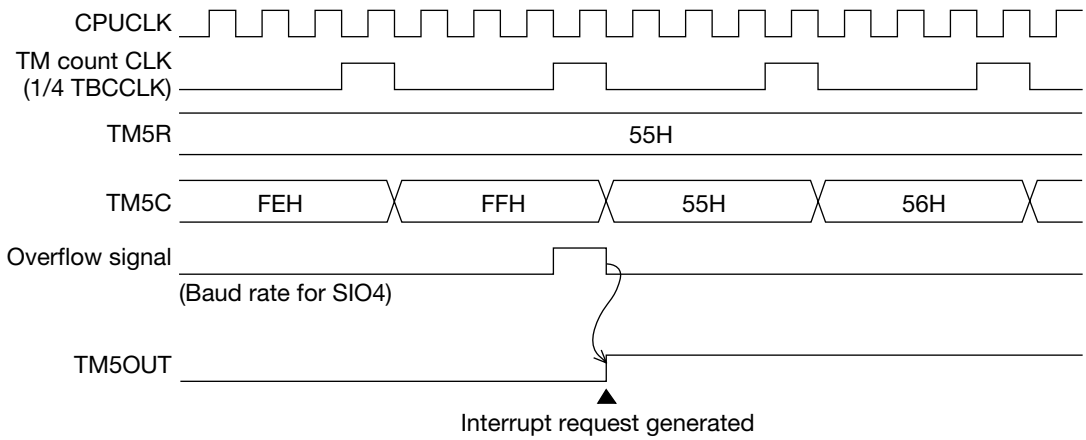


Figure 8-18 Timer 5 Operation Example

8.8.5 Timer 5 Interrupt

When a timer 5 interrupt factor occurs, the interrupt request flag (QTM5OV) is set to "1". The interrupt request flag (QTM5OV) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM5OV). The interrupt enable flag (ETM5OV) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0TM5OV and P1TM5OV). The interrupt priority setting flags are located in interrupt priority control register 6 (IP6).

Table 8-8 lists the vector address of the timer 5 interrupt factor and the interrupt processing flags.

Table 8-8 Timer 5 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 5	003A	QTM5OV	ETM5OV	P1TM5OV	P0TM5OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	17-15	17-20	17-28	

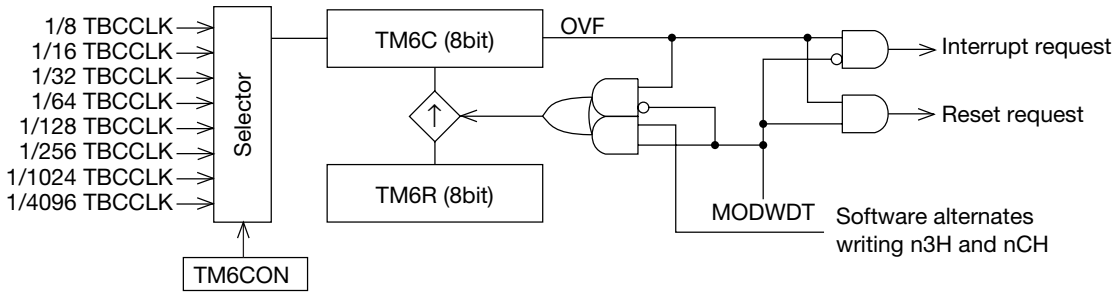
For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.9 Timer 6

Timer 6 is an 8-bit auto-reload timer that has two operating modes, auto-reload timer mode and watchdog timer (WDT) mode. If the counter overflows during the WDT mode, the system will be reset.

8.9.1 Timer 6 Configuration

Figure 8-19 shows the timer 5 configuration.



TM6C: General-purpose 8-bit timer 6 counter
TM6R: General-purpose 8-bit timer 6 register
TM6CON: General-purpose 8-bit timer 6 control register
MODWDT: WDT mode setting signal

Figure 8-19 Timer 6 Configuration

8.9.2 Description of Timer 6 Registers

(1) General-purpose 8-bit timer 6 counter (TM6C)

The general-purpose 8-bit timer 6 counter (TM6C) is an 8-bit up-counter.

- During auto-reload timer mode
  When an interrupt request is generated due to overflow of the counter, the contents of general-purpose 8-bit timer 6 register (TM6R) are loaded into TM6C.
- During WDT mode
  Counter overflow causes the system to be reset. When starting or initializing WDT, a special write operation to TM6C is necessary (so that WDT will not be easily initialized by an out-of-control program). The count value can be read during WDT operation, but once WDT is started, it is not possible to write to TM6C.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of TM6C are undefined.

[Note]
Writing a timer value to TM6C causes the same value to be also written to general-purpose 8-bit timer 6 register (TM6R).

(2) General-purpose 8-bit timer 6 register (TM6R)

The general-purpose 8-bit timer 6 register (TM6R) consists of 8 bits. This register stores the value to be reloaded into the general-purpose 8-bit timer 6 counter (TM6C).

During the auto-reload timer mode, the program can read from and write to TM6R. During the WDT mode, TM6R is read-only.

At reset (due to a $\overline{RES}$ input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of TM6R are undefined.

(3) General-purpose 8-bit timer 6 control register (TM6CON)

The general-purpose 8-bit timer 6 control register (TM6CON) consists of 7 bits.

During the auto-reload timer mode, the program can read from and write to TM6CON. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bit 4.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM6CON becomes 10H.
Figure 8-20 shows the TM6CON configuration.

[Description of each bit]

- WDTC0 to WDTC2 (bits 0 to 2)
  WDTC0 to WDTC2 specify the count clock for timer 6.

- ATMRUN (bit 3)
  During the auto-reload timer mode, ATMRUN specifies whether the count is running or halted.

  During the WDT mode, the value that has been written will be read.

- WDTRUN (bit 5)
  This read-only flag is read as "1" during counting in the WDT mode. With this flag, it is possible to determine whether the count operation in the WDT mode has started.

- WDTLDE (bit 6)
  During the WDT mode, WDT is initialized within a fixed period by loading the value of TM6R into TM6C. This load operation (WDT initialization) is performed by alternately writing "n3H" and "nCH" (where n is an arbitrary value from 0 to F) to TM6C.

  WDTLDE is a read-only flag used during initialization to determine whether the next value to be written to TM6C will be "n3H" or "nCH".

- MODWDT (bit 7)
  This bit specifies the timer 6 operating mode (auto-reload timer mode or WDT mode).

[Note]

Before setting MODWDT to "1" to enter the WDT mode, set the WDT overflow period with TM6C, TM6R and TM6CON (WDTC0 to WDTC2). It is not possible to modify the period once MODWDT is set to "1" and the WDT mode is entered. (Writes become invalid).

Since MODWDT is located within TM6CON, byte instructions can be used to simultaneously write to MODWDT and WDTC0 through WDTC2.

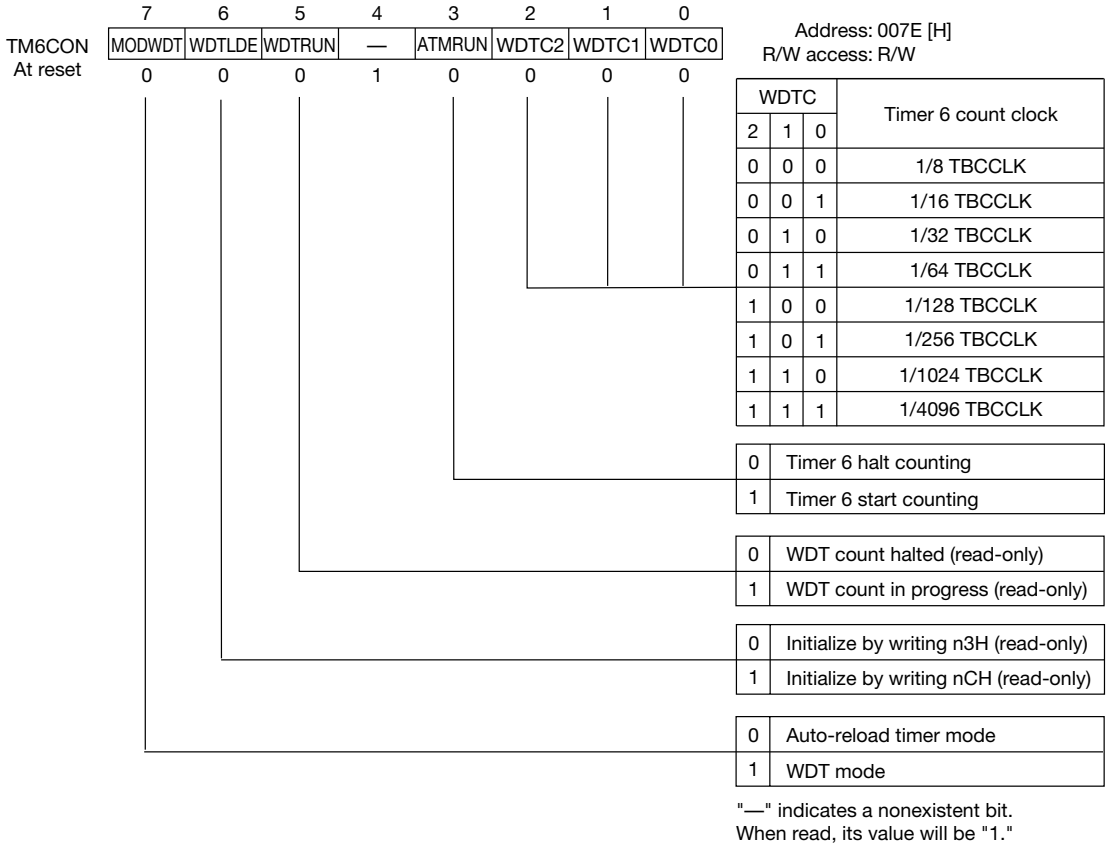


Figure 8-20 TM6CON Configuration

8.9.3 Example of Timer 6-related Register Settings

- **Auto-reload timer mode settings**

(1) General-purpose 8-bit timer 6 counter (TM6C)
Set the timer value that will be valid at the start of counting. When writing to TM6C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 6 register (TM6R).

(2) General-purpose 8-bit timer 6 register (TM6R)
This register sets the value to be loaded after general-purpose 8-bit timer 6 counter (TM6C) overflows. If the timer value (TM6C) and the reload value (TM6R) are identical, this register will automatically be set just by setting TM6C. If the values are different or are to be modified, this register must be set explicitly.

(3) General-purpose 8-bit timer 6 control register (TM6CON)
Bits 0 to 2 (WDTC0 to WDTC2) of this register specify the count clock for timer 6. If bit 3 (ATMRUN) is set to "1", timer 6 will begin counting. If reset to "0", timer 6 will halt counting.

- **Watchdog timer (WDT) mode settings**

(1) General-purpose 8-bit timer 6 register (TM6R)
This register sets the value to be loaded into general-purpose 8-bit timer 6 counter (TM6C).

(2) General-purpose 8-bit timer 6 control register (TM6CON)
(i) Specify the count clock for timer 6 with bits 0 to 2 (WDTC0 to WDTC2) of this register.

(ii) Set bit 7 (MODWDT) to "1" to enter the WDT mode.

(Settings (i) and (ii) can be performed simultaneously by using a byte instruction such as MOVB.)

(3) General-purpose 8-bit timer 6 counter (TM6C)
Write the WDT activation code, "n3H", to start WDT counting.

(At this time, the contents of TM6C are not modified. "n3H" is only used to activate WDT.)

Thereafter, WDT is initialized by alternately writing "nCH" and "n3H" before overflow. WDTLDE (bit 6) of TM6CON can be read to determine whether the value to be written for the next initialization is "nCH" or "n3H". "WDT initialization" is defined as loading the value of TM6R into TM6C. (n is an arbitrary value from 0 to F.)

8.9.4 Timer 6 Operation

- **Auto-reload timer mode**

When the MODWDT bit in TM6CON is reset to "0", the mode changes to the auto-reload timer mode. If the ATMRUN bit is set to "1", timer 6 will begin counting upward, running on the count clock selected by TM6CON. When TM6C overflows, an interrupt request is generated and the contents of TM6R are loaded into TM6C. This operation is repeated until the ATMRUN bit is reset to "0". Figure 8-21 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/8 TBCCLK).

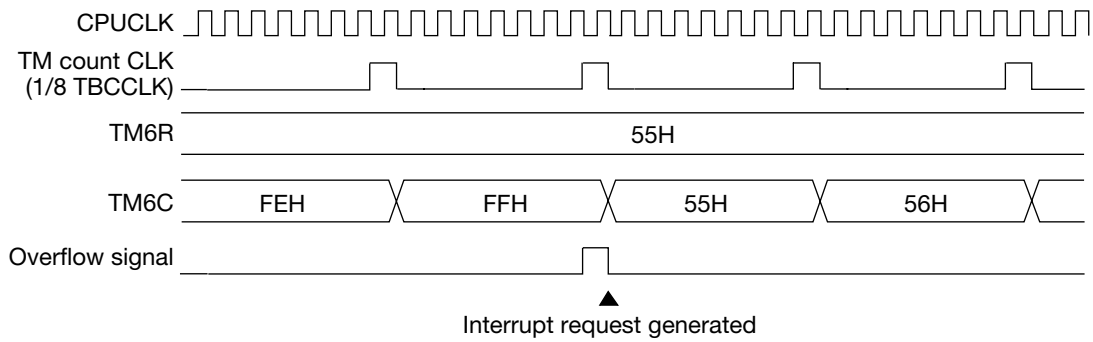


Figure 8-21 Timer 6 Operation (During Auto-Reload Timer Mode)

- **Watchdog timer (WDT) mode**

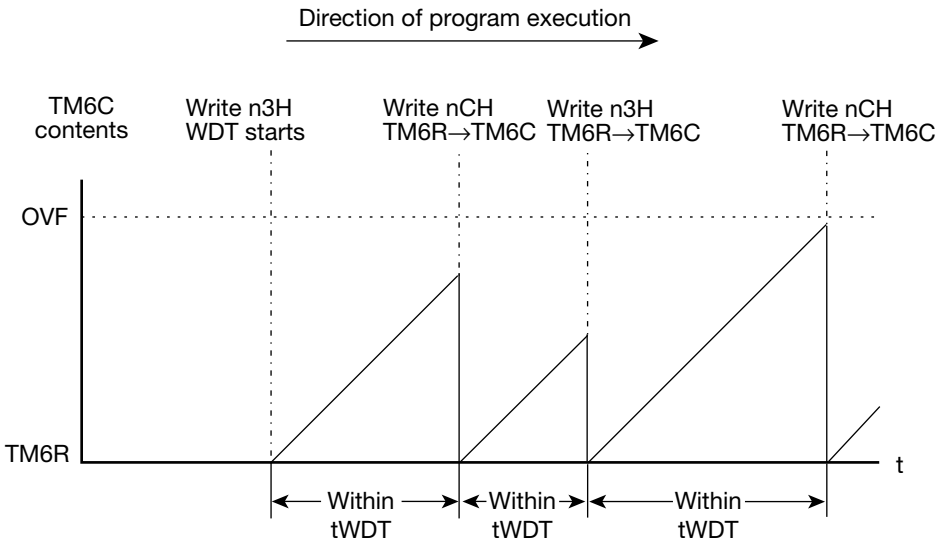
When the MODWDT bit in TM6CON is set to "1", the mode changes to the WDT mode. Once the WDT mode is set, it is not possible to return to the auto-reload timer mode until the system is reset. In the WDT mode, writing "n3H" to TM6C will cause the WDT count operation to begin. Thereafter, alternately writing "nCH" and "n3H" by the program will cause the contents of TM6R to be loaded into TM6C and initialize WDT.

If WDT initialization is not implemented within the fixed amount of time set by the count clock and the reload value, then TM6C will overflow and the system will be reset. To process a system reset, the branch address (2 bytes) stored in addresses 0004 to 0005 (vector address for reset by WDT) is loaded into the program counter.

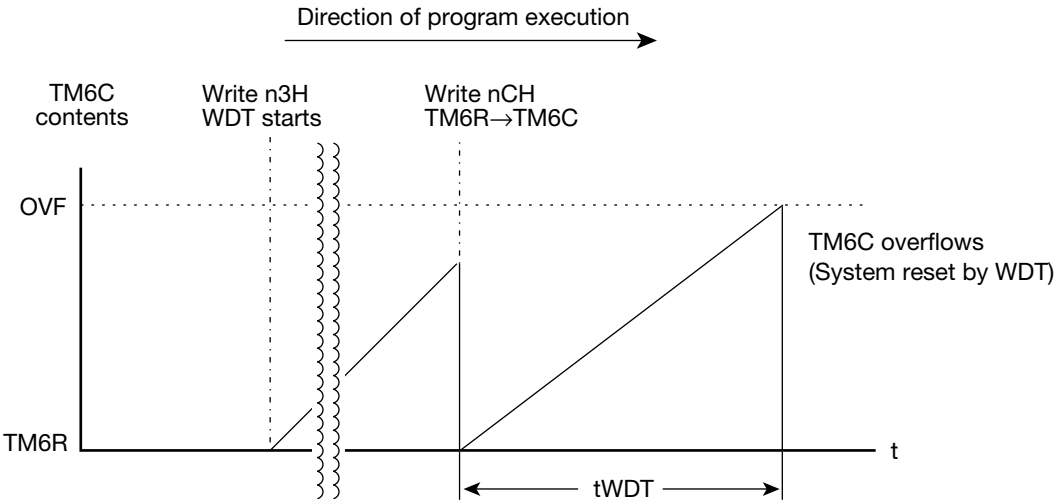
The time (tWDT) until TM6C overflows can be expressed by the below equation, where f [MHz] is the fundamental clock (CPUCLK), T is the TM6C count clock (divided value of TBCCLK), n is the divisor for the 1/n counter at the TBC front stage, and R is the value of TM6R.

$$t_{WDT} = (1/f) \times T \times n \times (256 - R) \quad [\mu s] \quad (R: 0 \text{ to } 255)$$

Figure 8-22 shows timing diagrams of an out-of-control program and detection by WDT. Figure 8-23 shows an example of an out-of-control program.

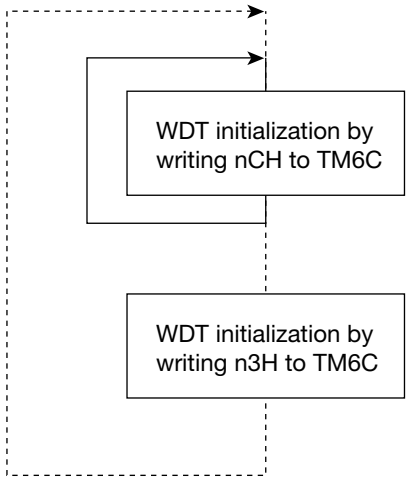


(a) Correct program execution

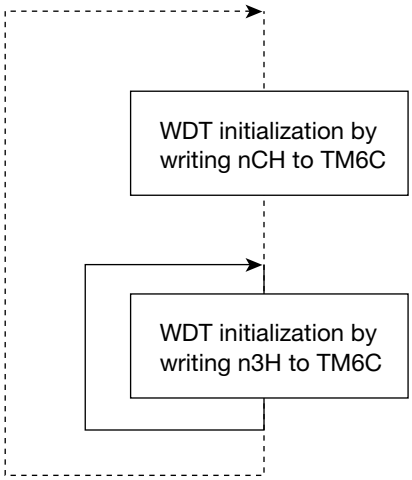


(b) Program runs out-of-control

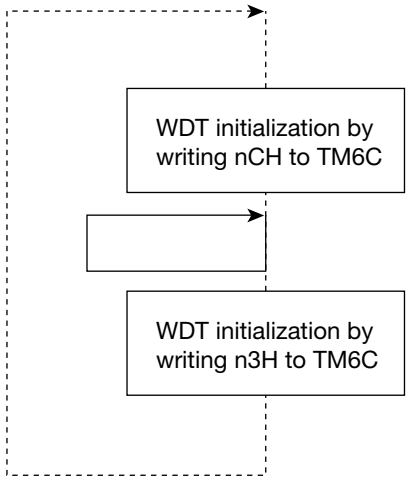
Figure 8-22 Timing Diagram of Out-of-Control Program Detection



(a) Execution of writing nCH only



(b) Execution of writing n3H only



(c) No writing to TM6C

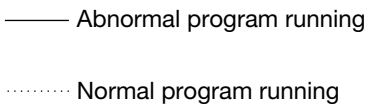


Figure 8-23 Example of Out-of-Control Program Detection

8.9.5 Timer 6 Interrupt (During Auto-Reload Timer Mode)

When a timer 6 interrupt factor occurs (during the auto-reload timer mode), the interrupt request flag (QTM6OV) is set to "1". The interrupt request flag (QTM6OV) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM6OV). The interrupt enable flag (ETM6OV) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0TM6OV and P1TM6OV). The interrupt priority setting flags are located in interrupt priority control register 7 (IP7).

Table 8-9 lists the vector address of the timer 6 interrupt factor and the interrupt processing flags.

Table 8-9 Timer 6 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 6	0042	QTM6OV	ETM6OV	P1TM6OV	P0TM6OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	17-15	17-20	17-29	

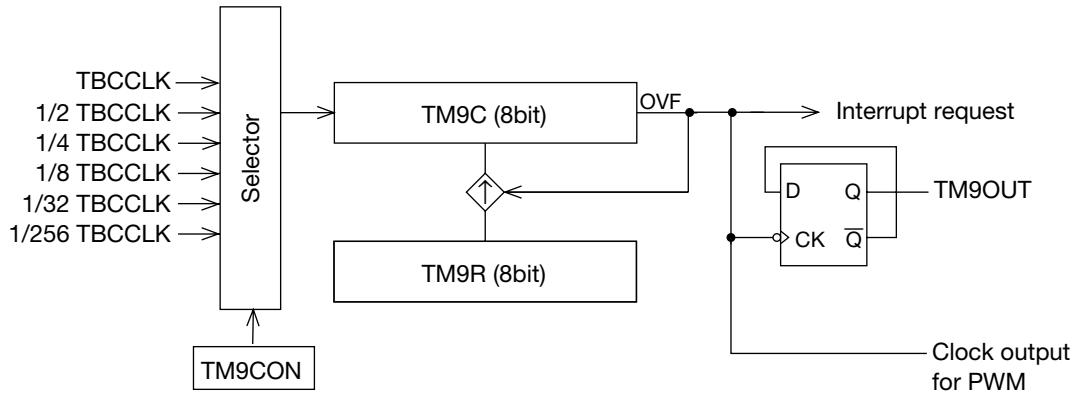
For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

8.10 Timer 9

Timer 9 is an 8-bit auto-reload timer that has the function for clock output for PWM. TM9OUT is not output on ports, but can be used by software as a flag.

8.10.1 Timer 9 Configuration

Figure 8-24 shows the timer 9 configuration.



TM9C: General-purpose 8-bit timer 9 counter
TM9R: General-purpose 8-bit timer 9 register
TM9CON: General-purpose 8-bit timer 9 control register

Figure 8-24 Timer 9 Configuration

8.10.2 Description of Timer 9 Registers

(1) General-purpose 8-bit timer 9 counter (TM9C)

The general-purpose 8-bit timer 9 counter (TM9C) is an 8-bit up-counter. When this counter overflows, an interrupt request is generated and it is loaded with the contents of general-purpose 8-bit timer 9 register (TM9R). This counter can also be used as a clock for PWM.

TM9C can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM9C are undefined.

[Note]
Writing a timer value to TM9C causes the same value to also be written to the general-purpose 8-bit timer 9 register (TM9R).

(2) General-purpose 8-bit timer 9 register (TM9R)

The general-purpose 8-bit timer 9 register (TM9R) consists of 8 bits. This register stores the value to be reloaded into the general-purpose 8-bit timer 9 counter (TM9C).

TM9R can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM9R are undefined.

(3) General-purpose 8-bit timer 9 control register (TM9CON)

The general-purpose 8-bit timer 9 control register (TM9CON) consists of 5 bits. Bits 0 to 2 (TM9C0 to TM9C2) of TM9CON select the timer 9 count clock, bit 3 (TM9RUN) specifies starting or halting the counting, and bit 7 (TM9OUT) specifies the initial timer output level (High or Low) at start-up. And each time TM0C overflows, the content of bit 7 (TM9OUT) is reversed.

TM9CON can be read from and written to by the program. However, write operations are invalid for bits 4 to 6. If read, a value of "1" will always be obtained for bits 4 to 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM9CON becomes 70H.

Figure 8-25 shows the TM9CON configuration.

[Note]
Just before TM9C overflows, if an SB, RB, XORB or other read-modify-write instruction is performed on TM9CON, then TM9OUT may not operate correctly.

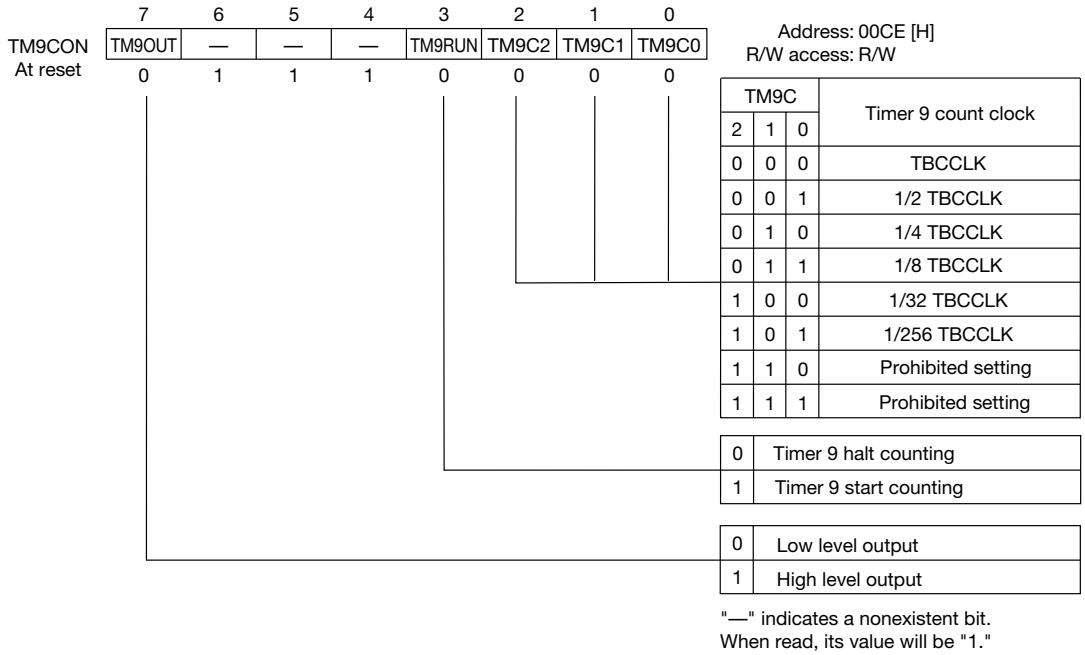


Figure 8-25 TM9CON Configuration

[Note]

Do not select a timer 9 count clock setting that is prohibited. If a "prohibited setting" is selected, timer 9 will not operate properly.

8.10.3 Example of Timer 9-related Register Settings

(1) General-purpose 8-bit timer 9 counter (TM9C)

Set the timer value that will be valid at the start of counting. When writing to TM9C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 9 register (TM9R).

(2) General-purpose 8-bit timer 9 register (TM9R)

This register sets the value to be loaded after general-purpose 8-bit timer 9 counter (TM9C) overflows. If the timer value (TM9C) and the reload value (TM9R) are identical, this register will automatically be set just by setting TM9C. If the values are different or are to be modified, this register must be set explicitly.

(3) General-purpose 8-bit timer 9 control register (TM9CON)

Bits 0 to 2 (TM9C0 to TM9C2) of this register specify the count clock for timer 9. If TM9OUT (timer output) is to be used, specify the initial value with bit 7 (TM9OUT). If bit 3 (TM9RUN) is set to "1", timer 9 will begin counting. If reset to "0", timer 9 will halt counting.

8.10.4 Timer 9 Operation

When the TM9RUN bit is set to "1", timer 9 will begin counting upward, running on the count clock selected by TM9CON. When TM9C overflows, an interrupt request is generated, the contents of TM9R are loaded into TM9C and the TM9OUT output is inverted. The initial value of the TM9OUT pin is specified by bit 7 (TM9OUT) of TM9CON. This operation is repeated until the TM9RUN bit is reset to "0". Overflow of TM9C can be used as the clock output for PWM. Figure 8-26 shows an operation example (for settings of 1/n counter frequency division ratio 1/1and 1/4 TBCCLK).

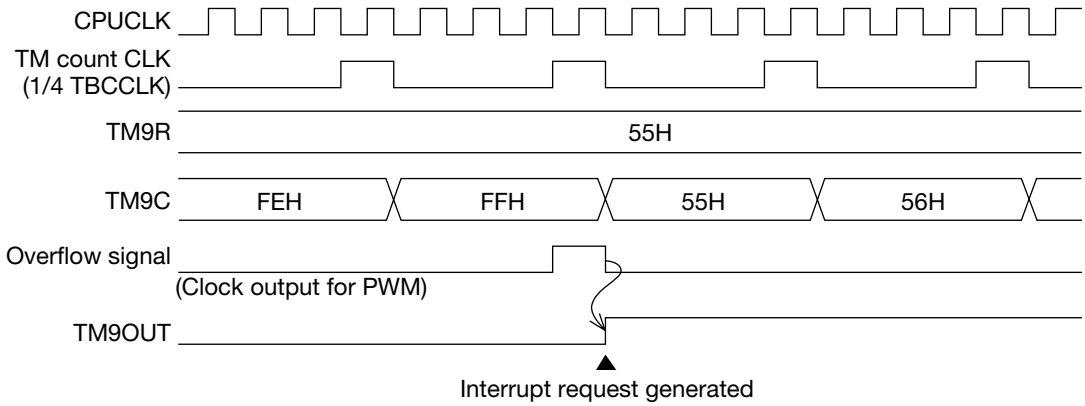


Figure 8-26 Timer 9 Operation Example

8.10.5 Timer 9 Interrupt

When a timer 9 interrupt factor occurs, the interrupt request flag (QTM9OV) is set to "1". The interrupt request flag (QTM9OV) is located in interrupt request register 4 (IRQ4).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM9OV). The interrupt enable flag (ETM9OV) is located in interrupt enable register 4 (IE4).

Three levels of priority can be set with the interrupt priority setting flags (P0TM9OV and P1TM9OV). The interrupt priority setting flags are located in interrupt priority control register 9 (IP9).

Table 8-10 lists the vector address of the timer 9 interrupt factor and the interrupt processing flags.

Table 8-10 Timer 9 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 9	0072	QTM9OV	ETM9OV	P1TM9OV	P0TM9OV
Symbols (byte) of registers that contain interrupt processing flags		IRQ4	IE4	IP9	
	Reference page	17-16	17-21	17-31	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 9

Capture/Compare Timer

9. Capture/Compare Timer

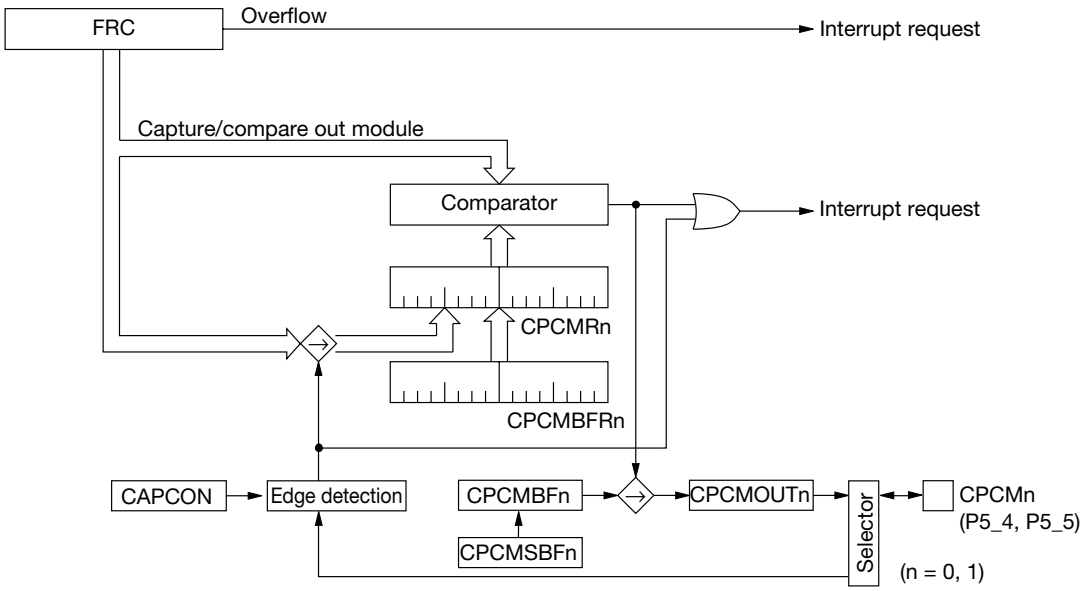
9.1 Overview

The MSM66577 family has a built-in 2-channel capture/compare timer.

Timer functions consist of a capture mode used for pulse width and cycle measurements and a compare out mode used for pulse output with real-time control. Functions can be selected for use with each of the two channels.

9.2 Capture/Compare Timer Configuration

Figure 9-1 shows the capture/compare timer configuration. The counter unit consists of a 16-bit free running counter (FRC) and two capture/compare out modules.



FRC: free running counter (16 bits)
CPCMRn: Capture/compare register (16 bits)
CAPCON: Capture control register
CPCMBFRn: Capture/compare buffer register (16 bits)
CPCMOUTn: Capture/compare out bit
CPCMn: Capture input/compare output pin (P5_4, P5_5)
CPCMBFn: Compare out buffer bit
CPCMSBFn: Compare out sub-buffer bit

Figure 9-1 Capture/Compare Timer Configuration

9.3 Capture/Compare Timer Registers

Table 9-1 lists a summary of SFRs for control of the capture/compare timer.

Table 9-1 Summary of SFRs for Control of the Capture/Compare Timer

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0040	Free running counter	—	FRC	R/W	16	0000	9-3
0041							
004A	Capture/compare register 0	—	CPCMR0	R/W	16	0000	9-6
004B							
004C	Capture/compare register 1	—	CPCMR1	R/W	16	0000	9-6
004D							
0050☆	Free running counter control register	FRCON	—	R/W	8	C0	9-4
0051☆	Capture control register	CAPCON	—	R/W	8	C0	9-7
0055☆	Compare control register 0	CPCMCON0	—	R/W	8	F8	9-6
0056☆	Compare control register 1	CPCMCON1	—	R/W	8	F8	9-6
00EA	Capture/compare buffer register 0	—	CPCMBFR0	R/W	16	0000	9-8
00EB							
00EC	Capture/compare buffer register 1	—	CPCMBFR1	R/W	16	0000	9-8
00ED							

[Notes]

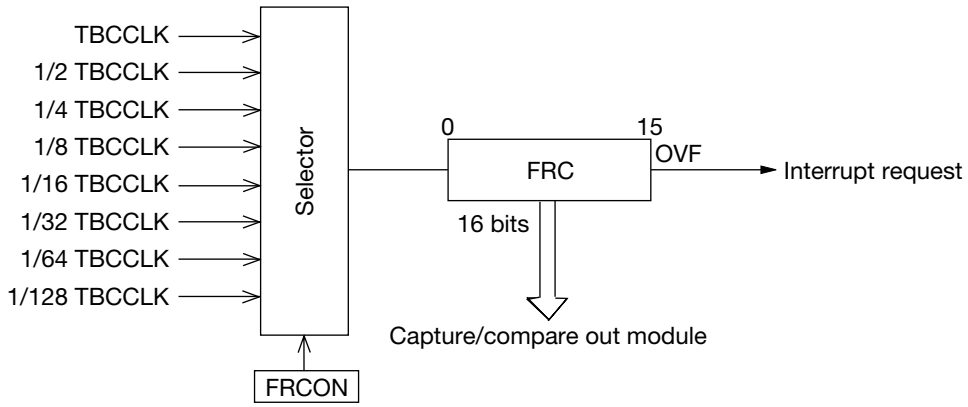
1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

9.4 16-Bit Free Running Counter (FRC)

A 16-bit free running counter (FRC) is used as the counter unit of the compare/capture timer.

9.4.1 16-Bit Free Running Counter Configuration

Figure 9-2 shows the 16-bit free running counter configuration.



FRC: free running counter (16 bits)
FRCON: free running counter control register

Figure 9-2 16-Bit Free Running Counter Configuration

9.4.2 Description of 16-bit Free Running Counter Register

(1) 16-bit free running counter (FRC)

The 16-bit free running counter (FRC) is a 16-bit up-counter. Counter overflow causes an interrupt request to be generated and the counter to be cleared to "0".

FRC can be read from and written to by the program.

Use the interrupt processing routine active when the interrupt request is generated to write the next values to the free running counter (FRC).

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), FRC becomes 0000H.

(2) Free running counter control register (FRCON)

The free running counter control register (FRCON) consists of 6 bits. FRCON selects the count clock for the free running counter (FRC), starts/stops the counter, and specifies the operating mode of the capture/compare out module. Bits 0 to 2 (FRCK0 to FRCK2) select the FRC count clock and bit 3 (FRRUN) specifies whether to run or stop the counter. Bit 4 (CP0MD) specifies the CPCM0 operating mode and bit 5 (CP1MD) specifies the CPCM1 operating mode.

FRCON can be read from and written to by the program. However, write operations are invalid for the upper 2 bits. If read, a value of "1" will always be obtained for the upper 2 bits.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), FRCON becomes C0H, TBCCLK is selected for the FRC count clock, and counting is halted. Also, the capture/compare out module will specify the compare out mode.

Figure 9-3 shows the FRCON configuration.

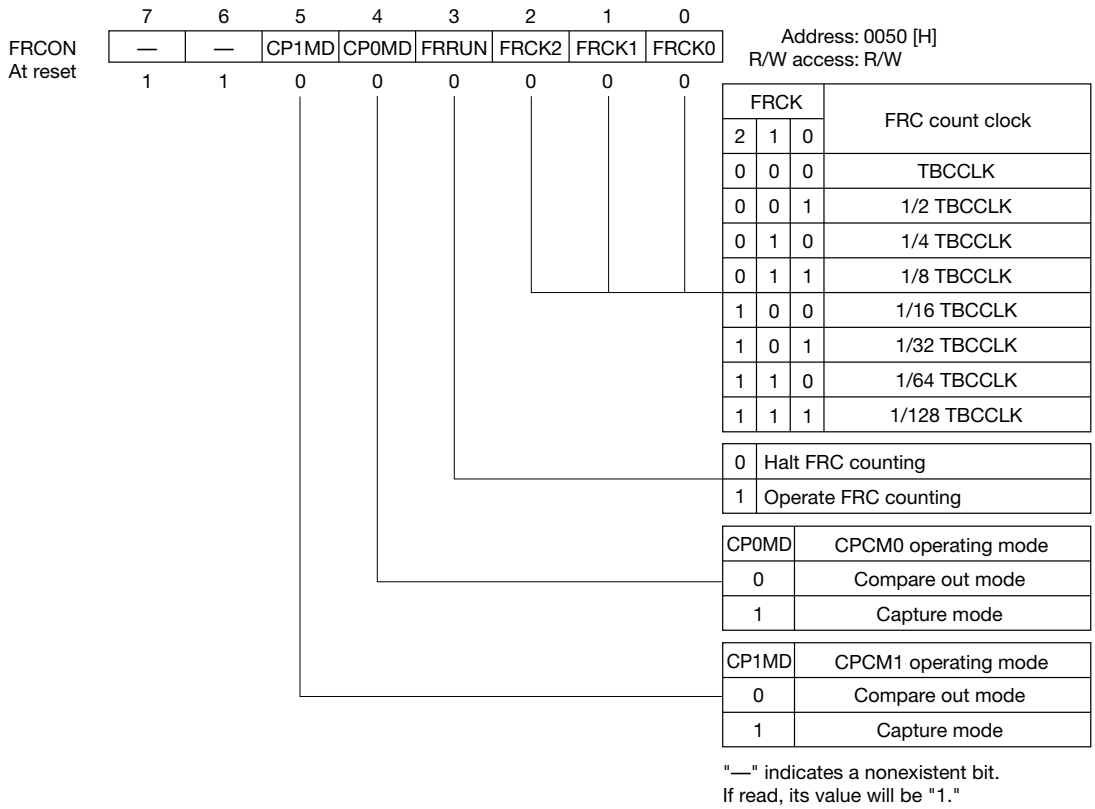


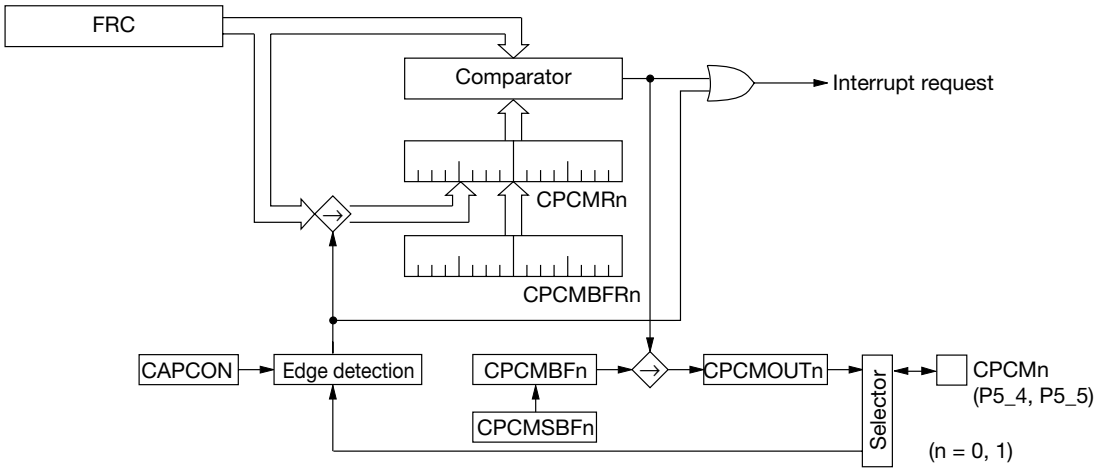
Figure 9-3 FRCON Configuration

9.5 Capture/Compare Out Modules

The MSM66577 family has two sets of capture/compare out modules. The configuration of the two sets is identical with the only difference being the address of registers in the SFR area.

9.5.1 Capture/Compare Out Module Configuration

Figure 9-4 shows the capture/compare out module configuration.



FRC: free running counter (16 bits)
CPCMRn: Capture/compare register (16 bits)
CAPCON: Capture control register
CPCMBFRn: Capture/compare buffer register (16 bits)
CPCMOUTn: Capture/compare out bit
CPCMn: Capture input/compare output pin (P5_4, P5_5)
CPCMBFn: Compare out buffer bit
CPCMSBFn: Compare out sub-buffer bit

Figure 9-4 Capture/Compare Out Module Configuration

9.5.2 Description of Capture/Compare Out Module Registers

(1) Capture/compare registers (CPCMR0, CPCMR1)

The capture/compare registers (CPCMR0 and CPCMR1) consist of 16 bits. In the compare out mode, CPCMR0 and CPCMR1 are always compared to the value of the free running counter (FRC). In the capture mode, when the edge specified as valid is input to a CPCMn pin, a capture event interrupt is generated, and at the same time, the contents of the free running counter (FRC) are loaded into CPCMR0 and CPCMR1.

CPCMR0 and CPCMR1 can be read from and written to by the program.

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CPCMR0 and CPCMR1 become 0000H.

(2) Capture/compare control registers (CPCMCON0, CPCMCON1)

The capture/compare control registers (CPCMCON0 and CPCMCON1) consist of 3 bits. In the compare out mode, if CPCMR0 and CPCMR1 match the value of the free running counter (FRC), the contents of CPCMBFn (bit 1) are loaded into CPCMOUTn (bit 0). At the same time, the contents of CPCMSBFn (bit 2) are loaded into CPCMBFn (bit 1). CPCMBFn is set to the level (High or Low level) that is desired at the time of the next match.

CPCMCON0 and CPCMCON1 can be read from and written to by the program. However, write operations are invalid for the upper 5 bits. If read, the value of the upper 5 bits is always "1".

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CPCMCON0 and CPCMCON1 become F8H.

Figure 9-5 shows the configuration of CPCMCON0 and CPCMCON1.

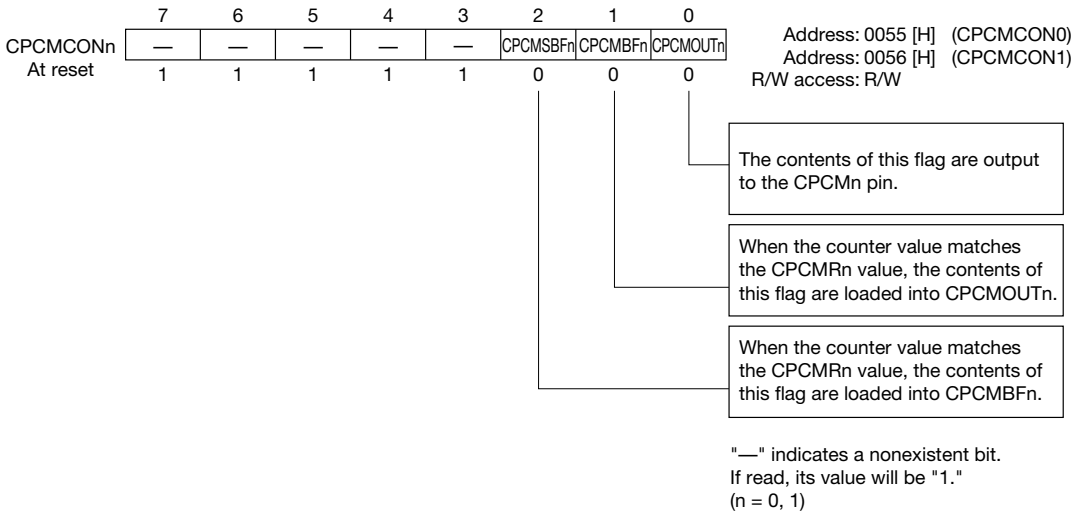


Figure 9-5 CPCMCON0 and CPCMCON1 Configuration

[Note]
Just before the occurrence of an event caused by compare out, if an SB, RB, XORB or other read-modify-write instruction is performed on CPCMCON0 or CPCMCON1, then CPCMBFn and CPCMOUTn may not operate correctly.

(3) Capture control register (CAPCON)

The capture control register (CAPCON) consists of 4 bits. When the capture/compare out module is in the capture mode, CAPCON specifies the valid edge for the signal input to CPCM0 and CPCM1 pins. Bits 2 and 3 (CP0E0 and CP0E1) specify the valid edge of the signal input to the CPCM0 pin, and bits 4 and 5 (CP1E0 and CP1E1) specify the valid edge of the signal input to the CPCM1 pin.

CAPCON can be read from and written to by the program. However, write operations are invalid for the upper 2 bits and lower 2 bits. If read, the upper 2 bits are always "1" and the lower 2 bits are always "0".

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CAPCON becomes C0H, and "capture input invalid" is specified for CPCM0 and CPCM1.

Figure 9-6 shows the configuration of CAPCON.

	7	6	5	4	3	2	1	0
CAPCON	—	—	CP1E1	CP1E0	CP0E1	CP0E0	—	—
At reset	1	1	0	0	0	0	0	0

Address: 0051 [H]
R/W access: R/W

CP0E		Valid edge during CPCM0 capture mode
1	0	
0	0	Input invalid
0	1	Falling edge
1	0	Rising edge
1	1	Both edges

CP1E		Valid edge during CPCM1 capture mode
1	0	
0	0	Input invalid
0	1	Falling edge
1	0	Rising edge
1	1	Both edges

"—" indicates a nonexistent bit.
If bits 6 and 7 are read, their value will be "1."
Bits 0 and 1 are read as "0."

Figure 9-6 CAPCON Configuration

[Note]
Set the minimum pulse width of the capture input longer than 1 CPU clock (CPUCLK). The capture input signal is sampled at the falling edge of the CPUCLK and used as the internal capture signal.

(4) Capture/compare buffer registers (CPCMBFR0, CPCMBFR1)

The capture/compare buffer registers (CPCMBFR0, CPCMBFR1) consist of 16 bits. During the compare out mode, if the value specified in CPCMR0 and CPCMR1 matches the value of the free running counter (FRC), the value set in CPCMBFR0 and CPCMBFR1 is loaded into CPCMR0 and CPCMR1.

The program can read from and write to CPCMBFR0 and CPCMBFR1.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CPCMBFR0 and CPCMBFR1 become 0000H.

9.6 Example of Capture/Compare Timer-related Register Settings

9.6.1 Capture Mode Settings

(1) Port 5 mode register (P5IO)

If CPCM0 is to be set to the capture mode, reset bit 4 (P5IO4) to "0" to configure the port as an input. If CPCM1 is to be set to the capture mode, reset bit 5 (P5IO5) to "0" to configure the port as an input.

(2) Port 5 secondary function control register (P5SF)

Bit 4 (P5SF4) specifies whether the CPCM0 capture input is pulled up. Bit 5 (P5SF5) specifies whether the CPCM1 capture input is pulled up.

(3) Capture control register (CAPCON)

Specify the valid edge for CPCM0 with bits 2 and 3 (CP0E0 and CP0E1). Specify the valid edge for CPCM1 with bits 4 and 5 (CP1E0 and CP1E1).

(4) Free running counter (FRC)

The initial value at the start of counting can be set by writing an arbitrary 16-bit value. FRC can be read from and written to during counting.

(5) Free running counter control register (FRCON)

Bits 0, 1, and 2 (FRCK0, FRCK1, and FRCK2) specify the count clock for the free running counter. To set CPCM0 to the capture mode, set bit 4 (CP0MD) to "1". To set CPCM1 to the capture mode, set bit 5 (CP1MD) to "1". If bit 3 (FRRUN) is set to "1", the free running counter will begin counting. If reset to "0", the free running counter will halt counting.

9.6.2 Compare Out Mode Settings

(1) Port 5 mode register (P5IO)

If CPCM0 is to be set to the compare out mode, set bit 4 (P5IO4) to "1" to configure the port as an output. If CPCM1 is to be set to the compare out mode, set bit 5 (P5IO5) to "1" to configure the port as an output.

(2) Port 5 secondary function control register (P5SF)

If CPCM0 is to be set to the compare out mode, set bit 4 (P5SF4) to "1" to configure the port as a secondary function output. If CPCM1 is to be set to the compare out mode, set bit 5 (P5SF5) to "1" to configure the port as a secondary function output.

(3) Capture/compare control registers (CPCMCON0, CPCMCON1)

If CPCM0 is to be set to the compare out mode, specify with bit 0 (CPCMOUT0) the initial value to be output to the CPCM0 pin, and specify with bit 1 (CPCMBF0) the value desired to be output from the CPCM0 pin when the value of the free running counter matches the contents of the CPCMR0. Specify with bit 2 (CPCMSBF0) the value desired to be output from the CPCM0 pin when the next value of the free running counter matches the contents of the CPCMR0. If CPCM1 is to be set to the compare out mode, specify with bit 0 (CPCMOUT1) the initial value to be output to the CPCM1 pin, and specify with bit 1 (CPCMBF1) the value desired to be output from the CPCM1 pin when the value of the free running counter matches the contents of the CPCMR1. Specify with bit 2 (CPCMSBF1) the value desired to be output from the CPCM1 pin when the next value of the free running counter matches the contents of the CPCMR1.

(4) Free running counter (FRC)

The initial value at the start of counting can be set by writing an arbitrary 16-bit value. FRC can be read from and written to during counting.

(5) Capture/compare registers (CPCMR0, CPCMR1)

If CPCM0 has been set to the compare out mode, set a count value in CPCMR0 at which to change the CPCM0 pin output. If CPCM1 has been set to the compare out mode, set a count value in CPCMR1 at which to change the CPCM1 pin output.

(6) Capture/compare buffer registers (CPCMBFR0, CPCMBFR1)

If CPCM0 has been set to the compare out mode, CPCMBF0specifies the next count value of CPCMR0 at which output of the CPCM0 pin will change. If CPCM1 has been set to the compare out mode, CPCMBF1 specifies the next count value of CPCMR1 at which output of the CPCM1 pin will change.

(7) Free running counter control register (FRCON)

Bits 0, 1, and 2 (FRCK0, FRCK1, and FRCK2) specify the count clock for the free running counter. To set CPCM0 to the compare out mode, reset bit 4 (CP0MD) to "0". To set CPCM1 to the compare out mode, reset bit 5 (CP1MD) to "0". If bit 3 (FRRUN) is set to "1", the free running counter will begin counting. If reset to "0", the free running counter will halt counting.

9.7 Capture/Compare Timer Operation

9.7.1 Capture Mode Operation

When the free running counter (FRC) is in the RUN state, if the valid edge specified by CAPCON is input to the CPCM0 or CPCM1 pins, the capture event will generate an interrupt request, and the contents of the free running counter (FRC) will be simultaneously loaded into CPCMRn (where n = 0, 1).

Figure 9-7 shows an operation example of the capture mode.

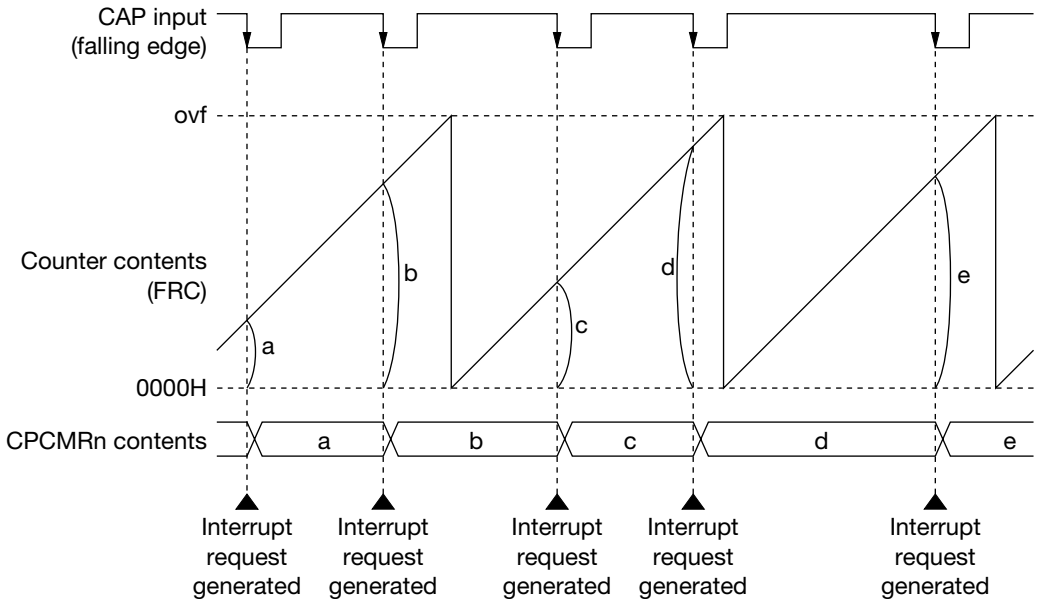


Figure 9-7 Capture Module Operation Example

[Note]
Set the minimum pulse width of the capture input to at least 1 CPU clock (CPUCLK). The capture input signal is sampled at the falling edge of the CPUCLK and used as the internal capture signal.

9.7.2 Compare Out Mode Operation

When the free running counter (FRC) is in the RUN state, CPCMR0 and CPCMR1 are always compared to the value of the free running counter (FRC). If they match, an interrupt request is generated by compare out and the contents of the CPCMBFn (bit 1) of CPCMCON0 and CPCMCON1 are loaded into CPCMOUTn (bit 0) (where n = 0, 1).Also, the contents of the CPCMSBFn (bit s) are loaded into CPCMBFn (bit 1) and the contents of the CPCMBRn are loaded into CPCMRn. Therefore, set CPCMRn with the level desired at the timing for the next match and set CPCMBFn with the level desired at the timing for the match following the next match (where n = 0, 1). Use the interrupt processing routine active when the interrupt request is generated to write the next values to CPCMRn and CPCMBFn (where n = 0, 1).

Figure 9-8 shows an example of compare out mode operation.

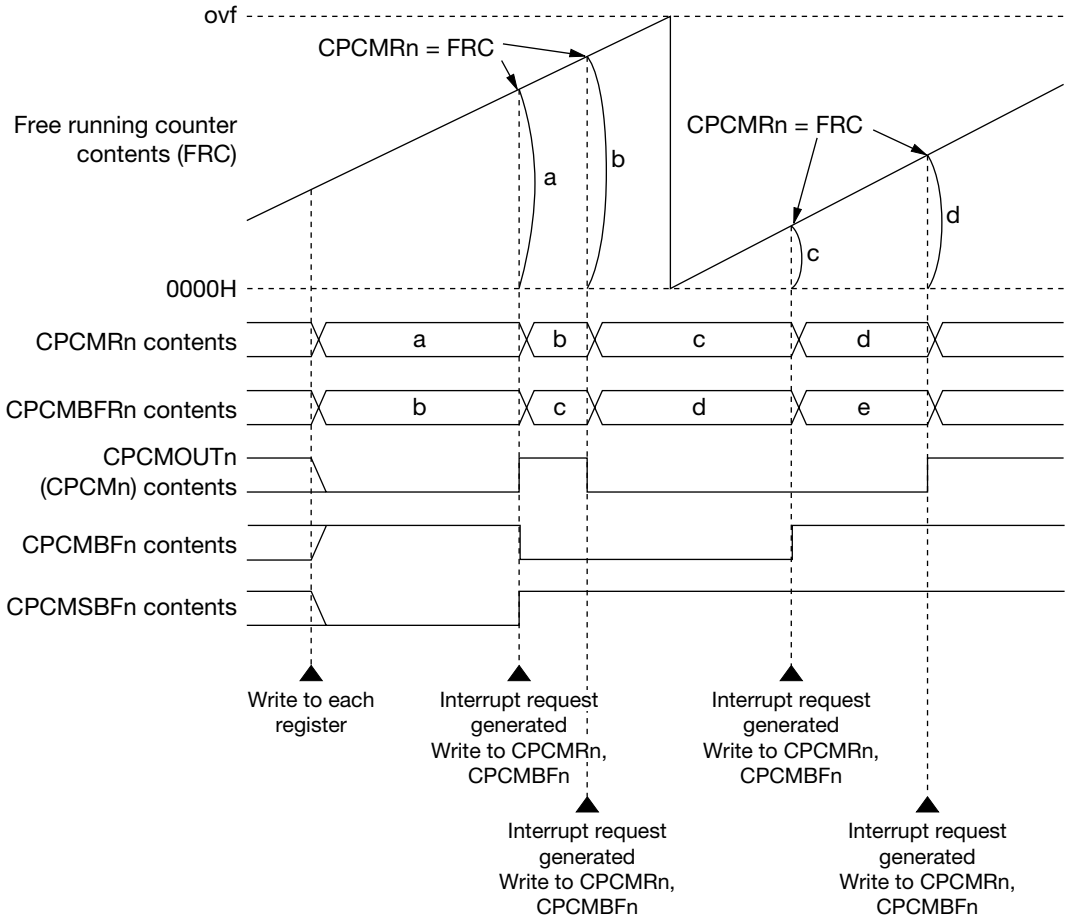


Figure 9-8 Compare Out Mode Operation Example

9.8 Example Timings for Changing the Output Level of Compare Out

Example timings in the compare out mode for changing the output level of the CMP output are shown below.

Figure 9-9 shows an example when 1/4 CPUCLK is selected as the clock source for the free running counter (FRC). When the contents of CPCMRn are 100H, the FRC and CPCMRn matching signal will be HIGH 1 CLK after the interval where FRC is 100H. The CPCMn output pin (CPCMOUTn) changes at the falling edge of the logical AND of this matching signal with the FRC clock pulse. Further, the corresponding interrupt request flag is set at the next M1S1 (signal that indicates the beginning of an instruction).

This example shows the timing of an output level change when 1/2 CPUCLK or larger frequency division ratio is selected as the FRC clock source.

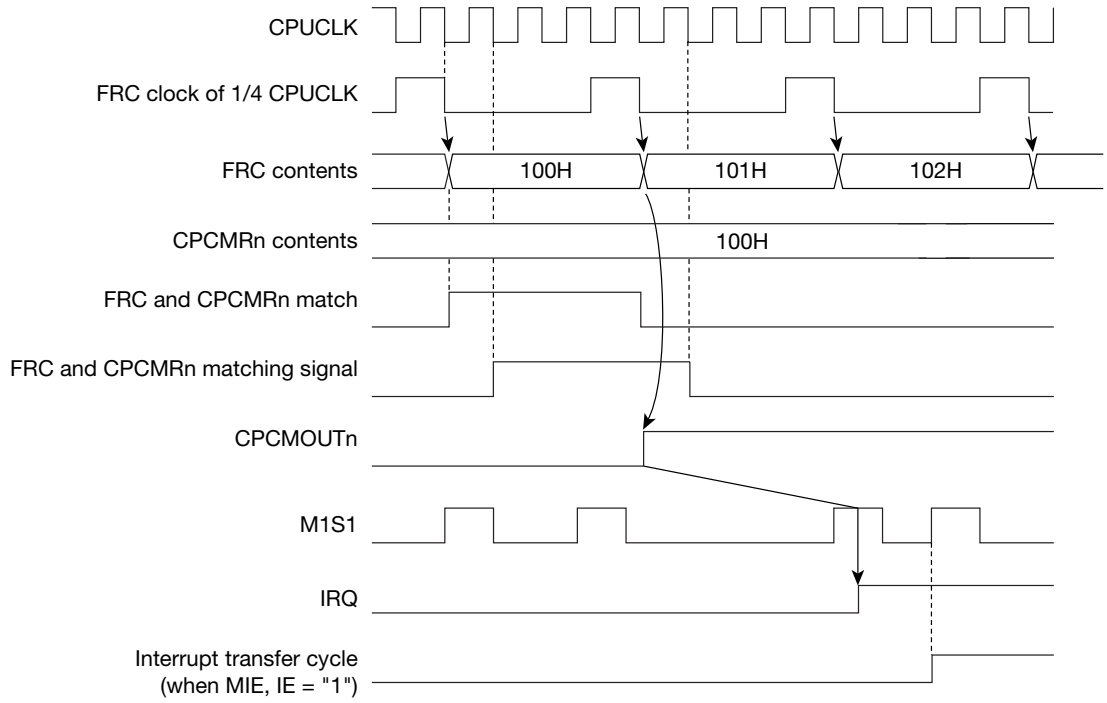


Figure 9-9 Example Timing for Changing the Output Level of Compare Out (FRC Clock = 1/4 CPUCLK)

Figure 9-10 shows an example when 1/1 CPUCLK is selected as the clock source for the free running counter (FRC). When the contents of CPCMRn are 100H, the FRC and CPCMRn matching signal will be HIGH 1 CLK after the interval where FRC is 100H. The CPCMn output pin (CPCMOUTn) changes at the falling edge of the logical AND of this matching signal with the FRC clock pulse. Therefore, when the FRC clock is set as 1/1 CPUCLK, the timing at which the CPCM output pin will change is FRC = CPCMRn + 01H (when FRC is 101H in the figure below). The corresponding interrupt request flag is set at the next M1S1 (signal that indicates the beginning of an instruction).

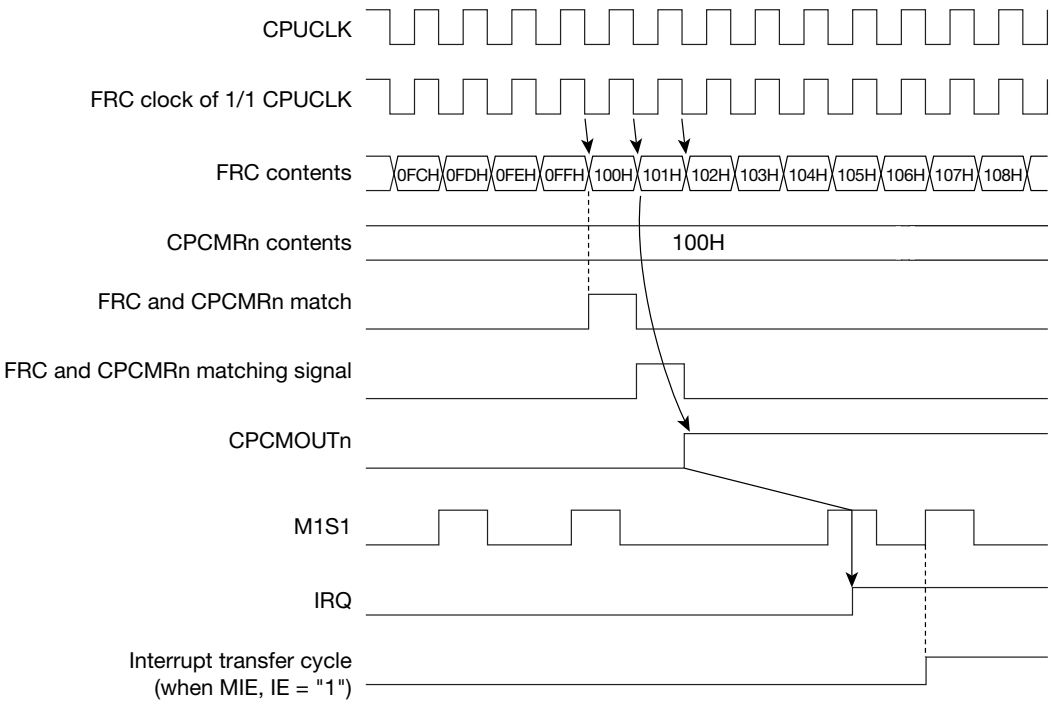


Figure 9-10 Example Timing for Changing the Output Level of Compare Out (FRC Clock = 1/1 CPUCLK)

[Note]

In the above, the free running counter (FRC) clock is described as the CPUCLK base. However, the actual FRC clock source is the time base counter (TBC) output. Therefore, the example of Figure 9-10 is limited to the case where TBCCLK is selected as the FRC clock with TBCCLK = CPUCLK (when the 1/n counter at the TBC front stage is set with the value 1/1).

For further details regarding TBC, refer to Chapter 7, "Time Base Counter (TBC)".

9.9 Capture/Compare Timer Interrupt

When each capture/compare timer interrupt factor occurs, the corresponding interrupt request flag is set to "1". Interrupt request flags are located in interrupt request register 0 (IRQ0).

Interrupts can be enabled or disabled by interrupt enable flags that correspond to each interrupt factor. The interrupt enable flags are located in interrupt enable register 0 (IE0).

Corresponding to each interrupt factor, interrupt priority setting flags can set three levels of priority for each interrupt factor. The interrupt priority setting flags are located in interrupt priority control registers 0 and 1 (IP0 and IP1).

Table 9-2 lists the vector address of each interrupt factor of the capture/compare timer and the interrupt processing flags.

Table 9-2 Capture/Compare Timer Vector Addresses and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of free running counter	000C	QFRCOV	EFRCOV	P1FRCOV	P0FRCOV
CPCM0 capture input	0016	QCPCM0	ECPCM0	P1CPCM0	P0CPCM0
CPCM0 compare match					
CPCM1 capture input	0018	QCPCM1	ECPCM1	P1CPCM1	P0CPCM1
CPCM1 compare match					
Symbols (BYTE) of registers that contain interrupt processing flags		IRQ0	IE0	IP0/IP1	
	Reference page	17-12	17-17	17-22/17-23	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 10

Real-Time Counter (RTC)

10. Real-Time Counter (RTC)

10.1 Overview

The MSM66577 family contains one internal 15-bit real-time clock counter (RTC).

The real-time counter runs on the clock obtained from the oscillation circuit (32.768 kHz) connected to the XT pins. Interrupt requests at 1, 0.5, 0.25, and 0.125 seconds can be obtained from the output of the real-time counter.

Counting continues even when in a standby state (STOP, HALT and HOLD modes). The STOP and HALT modes can be released by the real-time counter interrupt.

10.2 Real-Time Counter Configuration

Figure 10-1 shows the real-time counter configuration.

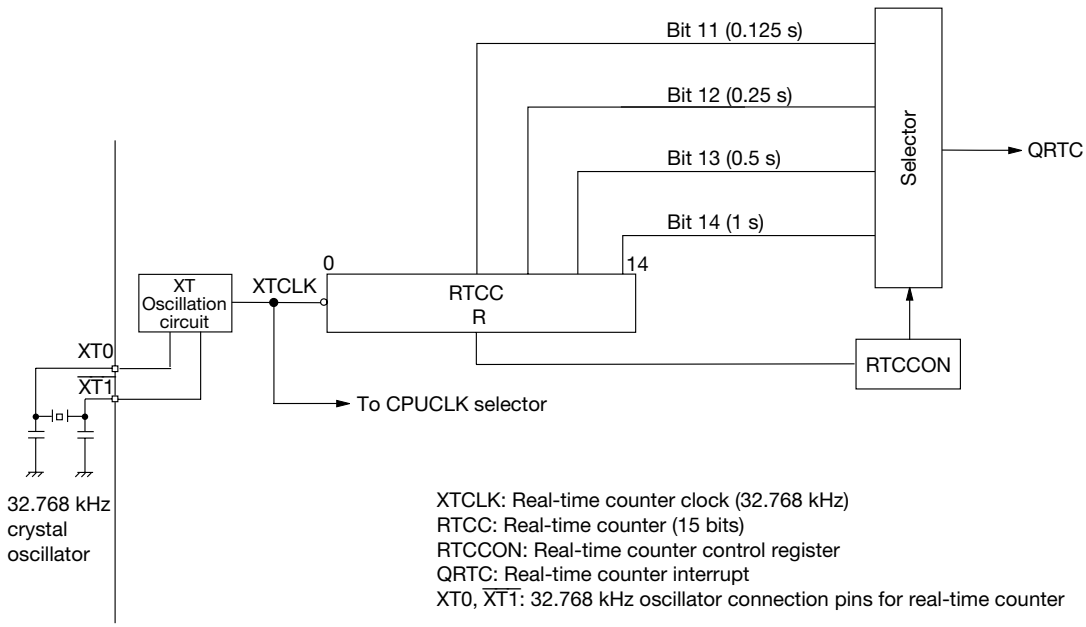


Figure 10-1 Real-Time Counter Configuration

10.3 Real-Time Counter Control Register (RTCCON)

The real-time clock control register (RTCCON) consists of 4 bits. All real-time counter settings are performed with RTCCON.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), RTCCON becomes F0H.

Figure 10-2 shows the RTCCON configuration.

[Description of each bit]

- RTCCL (bit 0)
  This bit is used to reset the real-time counter. If RTCCL is set to "1", the real-time counter will be reset to "0". When the real-time counter is reset, RTCCL is also simultaneously reset to "0".

- RTCIE (bit 1)
  This bit enables or disables the real-time counter interrupt.

- SELRTI0, SELRTI1 (bits 2 and 3)
  These bits select the interrupt cycle for the real-time counter interrupt.

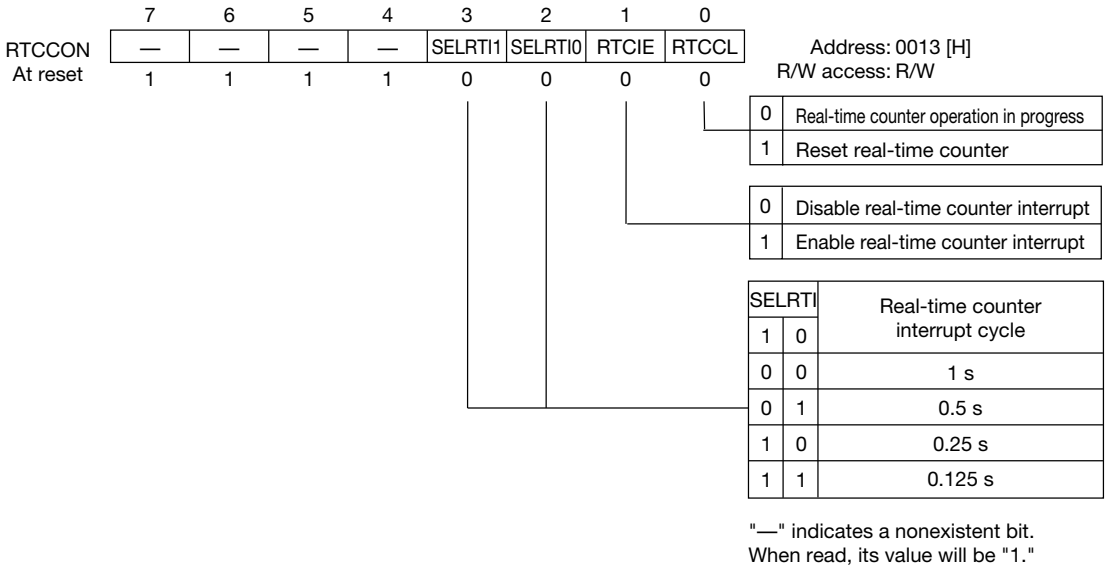


Figure 10-2 RTCCON Configuration

10.4 Example of Real-Time Counter Register Settings

(1) Peripheral Control Register (PRPHCON)
If the real-time counter related is to run on an external clock input to the XT0 pin (instead of the clock from the XT oscillation circuit), set bit 4 (EXTXT) to "1". Thereafter, an external clock can be input to the XT0 pin.

(2) Real-Time Counter Control Register (RTCCON)
To reset the real-time counter, set bit 0 (RTCCL) to "1". If the real-time interrupt is to be used, specify the real-time counter interrupt request cycle with bits 2 and 3 (SELRTI0 and SELRTI1) and set bit 1 (RTCIE) to "1" to enable the interrupt.

10.5 Real-Time Counter Operation

The real-time counter counts upward at the falling edge of XTCLK (XT clock). The output of real-time counter bits 11 through 14, as selected by bits 2 and 3 (SELRTI0 and SELRTI1) of the real-time counter control register (RTCCON), generate real-time counter interrupt requests.

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the real-time counter is reset to "0". The real-time counter can also be reset to "0" by setting the RTCCL bit of RTCCON to "1".

10.6 Real-Time Counter Interrupt

When the real-time counter interrupt factor occurs, the interrupt request flag (QRTC) is set to "1". The interrupt request flag (QRTC) is located in interrupt request register 3 (IRQ3).

Interrupt can be enabled or disabled by the interrupt enable flag (ERTC). The interrupt enable flag (ERTC) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0RTC and P1RTC). The interrupt priority setting flags are located in interrupt priority control register 7 (IP7).

Table 10-1 lists the vector address of the real-time counter interrupt factor and the interrupt processing flags.

Table 10-1 Real-Time Counter Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Real-time counter output (Cycle: 0.125 to 1 s)	0048	QRTC	ERTC	P1RTC	P0RTC
Symbols (BYTE) of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	17-15	17-20	17-29	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 11

PWM Function

11. PWM Function

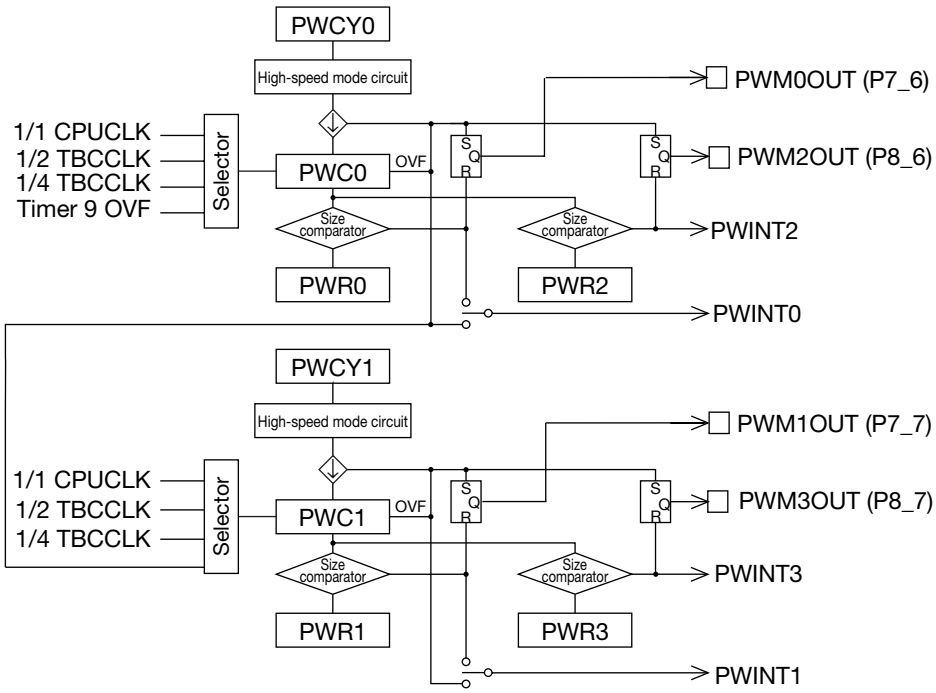
11.1 Overview

The MSM66577 family contains 4 channels of PWM (Pulse Width Modulation) function that can vary the duty with a fixed cycle. The resolution of each channel of PWM output is 8 bits. Use of this function as 2 channels of PWM with 16-bit resolution is also possible. When used as a 16-bit PWM, a high-speed mode is available that does not degrade the resolution of PWM output.

11.2 PWM Configuration

The MSM66577 family has two sets of 2-channel 8-bit PWMs (8-bit PWM0 and 8-bit PWM1) that share a common counter. These can be cascaded and used as 16-bit PWM (16-bit mode).

Figure 11-1 shows the PWM configuration.



PWCY0, PWCY1: PWM cycle register (8 bits)
PWC0, PWC1: PWM counter (8 bits)
PWR0 to PWR3: PWM register (8 bits)
PWM0OUT to PWM3OUT: PWM output pin
PWMINT0 to PWMINT3: Interrupt request

Figure 11-1 PWM Configuration

11.3 PWM Register

Table 11-1 lists a summary of SFRs for PWM control.

Table 11-1 Summary of SFRs for PWM Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0090	PWM register 0	PWR0	PWR01	R/W	8/16	00	11-4
0091	PWM register 1	PWR1				00	
0092	PWM register 2	PWR2	PWR23	R/W	8/16	00	11-4
0093	PWM register 3	PWR3				00	
0094	PWM cycle register 0	PWCY0	PWCY	R/W	8/16	00	11-3
0095	PWM cycle register 1	PWCY1				00	
0096	PWM counter 0	PWC0	PWC	R/W	8/16	00	11-3
0097	PWM counter 1	PWC1				00	
0098	PWM control register 0	PWCON0	—	R/W	8	00	11-4
0099☆	PWM control register 1	PWCON1	—	R/W	8	FE	11-6

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

11.3.1 Description of PWM Registers

(1) PWM counters (PWC0, PWC1)

The PWM counters (PWC0, PWC1) are 8-bit up-counters. When overflow occurs, the value in PWM cycle registers (PWCY0, PWCYC1) is loaded into PWC0 and PWC1.

PWC0 and PWC1 can be read from and written to by the program. PWC0 and PWC1 can also be accessed as 16-bit PWC. During a 16-bit access, PWC1 is the upper 8 bits and PWC0 is the lower 8 bits of PWC.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWC0 and PWC1 become 00H.

[Note]

Writing a count value to PWC0 causes the same value to also be written to PWM cycle register 0 (PWCY0). Similarly, writing a count value to PWC1 causes the same value to also be written to PWM cycle register 1 (PWCY1).

(2) PWM cycle registers (PWCY0, PWCY1)

The PWM cycle registers (PWCY0, PWCY1) are 8-bit registers that set the PWM cycle.

PWCY0 and PWCY1 can be read from and written to by the program. PWCY0 and PWCY1 can also be accessed as 16-bit PWCY. During a 16-bit access, PWCY1 is the upper 8 bits and PWCY0 is the lower 8 bits of PWCY.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWCY0 and PWCY1 become 00H.

[Note]

The cycle set in PWCY0 must be longer than the duty value set by PWR0 and PWR2. Also, the cycle set in PWCY1 must be longer than the duty value set by PWR1 and PWR3. During the 16-bit mode, the cycle set in PWCY must be longer than the duty value set in PWR01 and PWR23.

(3) PWM registers (PWR0 to PWR3)

The PWM registers (PWR0 to PWR3) are 8 bit registers that set the duty value. The duty value setting for PWR0 and PWR2 is limited to within the cycle range set by PWCY0. Also, the duty value setting for PWR1 and PWR3 is limited to within the cycle range set by PWCY1.

PWR0 to PWR3 can be read from and written to by the program. PWR0 and PWR1 can also be accessed as the 16-bit PWR01. PWR2 and PWR3 can also be accessed as the 16-bit PWR23. During a 16-bit access, PWR1 is the upper 8 bits and PWR0 is the lower 8 bits of PWR01, and PWR3 is the upper 8 bits and PWR2 is the lower 8 bits of PWR23.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWR0 to PWR3 become 00H.

[Note]
During the 16-bit mode, the duty valu set by PWR01 and PWR23 is limited to within the cycle range set by PWCY.

(4) PWM control register 0 (PWCON0)

The PWM control register 0 (PWCON0) consists of 8 bits. PWCON0 starts and stops the PWM counters (PWC0, PWC1), selects the counter clock, and specifies the interrupt factor of PWINT0 and PWINT1.

PWCON0 can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWCON0 becomes 00H.

Figure 11-2 shows the PWCON0 configuration.

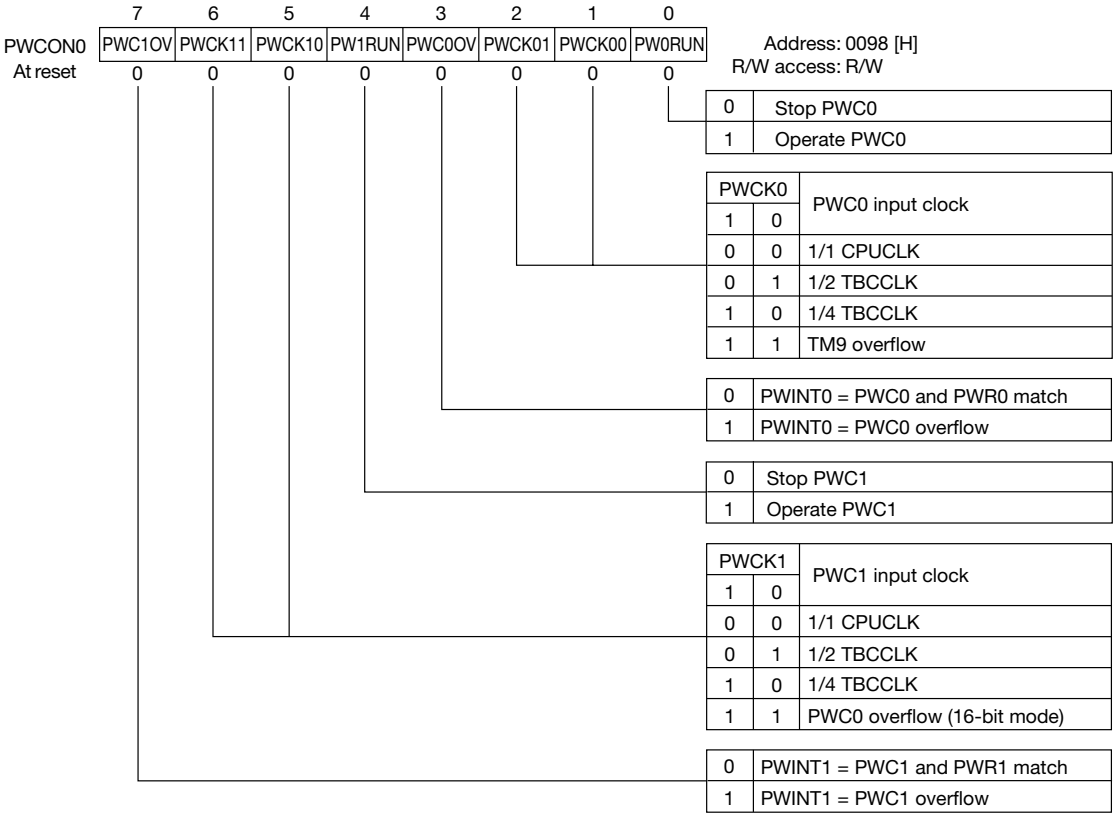


Figure 11-2 PWCON0 Configuration

(5) PWM control register 1 (PWCON1)

The PWM control register 1 (PWCON1) consists of 1 bit. PWCON1 register is used to select normal mode or high-speed mode of PWM. If bit 0 (PWHSM) is set to "1", the mode changes to high-speed mode. High-speed mode can only be used during the 16-bit mode.

PWCON can be read from or written to by the program. However, write operations are invalid for bits 1 through 7. If read, a value of "1" will always be obtained for bits 1 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWCON1 becomes FEH.

Figure 11-3 shows the PWCON1 configuration.

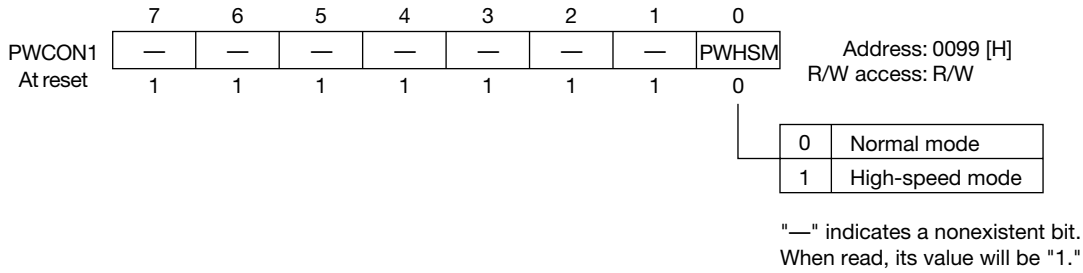


Figure 11-3 PWCON1 Configuration

[Note]
High-speed mode is only valid when 16-bit PWM is used.

11.3.2 Example of PWM-related Register Settings

- **8-bit PWM settings**

(1) Port 7 mode register (P7IO)
If PWM0OUT is to be used, set bit 6 (P7IO6) to "1" to configure the port as an output. If PWM1OUT is to be used, set bit 7 (P7IO7) to "1" to configure the port as an output.

(2) Port 8 mode register (P8IO)
If PWM2OUT is to be used, set bit 6 (P8IO6) to "1" to configure the port as an output. If PWM3OUT is to be used, set bit 7 (P8IO7) to "1" to configure the port as an output.

(3) Port 7 secondary function control register (P7SF)
If PWM0OUT is to be used, set bit 6 (P7SF6) to "1" to configure the port as a secondary function output. If PWM1OUT is to be used, set bit 7 (P7SF7) to "1" to configure the port as a secondary function output. In the PWM initial state or when PWM function is halted, the PWM output port is fixed at "1". To fix the port at "0", return the port to a primary function.

(4) Port 8 secondary function control register (P8SF)
If PWM2OUT is to be used, set bit 6 (P8SF6) to "1" to configure the port as a secondary function output. If PWM3OUT is to be used, set bit 7 (P8SF7) to "1" to configure the port as a secondary function output. In the PWM initial state or when PWM function is halted, the PWM output port is fixed at "1". To fix the port at "0", return the port to a primary function.

(5) PWM counters (PWC0, PWC1)
Set these counters with the value at which to start counting. Writing to PWC0 and PWC1 causes the same value to be simultaneously and automatically written to PWCY0 and PWCY1.

(6) PWM cycle registers (PWCY0, PWCY1)
If PWM0OUT and PWM2OUT are to be used, set the PWM cycle in PWCY0. If PWM1OUT and PWM3OUT are to be used, set the PWM cycle in PWCY1.

(7) PWM registers (PWR0 to PWR3)
If PWMnOUT are to be used, set the desired output duty value in PWRn (where n = 0 to 3). Set a value for PWR0 and PWR2 that is larger than the value of PWCY0. Set a value for PWR1 and PWR3 that is larger than the value of PWCY1.

(8) PWM control register 0 (PWCON0)

If PWM0OUT and PWM2OUT are to be used, set the count clock for PWM counter 0 (PWC0) with bits 1 and 2 (PWCK00, PWCK01), and specify the interrupt factor that will initiate a PWINT0 interrupt request with bit 3 (PWC0OV). If bit 0 (PW0RUN) is set to "1", the PWM counter 0 (PWC0) begins counting. If reset to "0" the counting is halted.

If PWM1OUT and PWM3OUT are to be used, set the count clock for PWM counter 1 (PWC1) with bits 5 and 6 (PWCK10, PWCK11), and specify the interrupt factor that will initiate a PWINT1 interrupt request with bit 7 (PWC1OV). If bit 4 (PW1RUN) is set to "1", the PWM counter 1 (PWC1) begins counting. If reset to "0" the counting is halted.

[Equation to Calculate 8-Bit PWM Cycle]

$f_{(PWM8)} = PWCLK / (256 - PWCYn)$

$f_{(PWM8)}$: PWM cycle [Hz]
PWCLK : PWM input clock frequency [Hz]
PWCYn : Value of PWCY0 or PWCY1 (8 bits)

- **16-bit PWM settings**

(1) Port 7 mode register (P7IO)

If PWM1OUT is to be used, set bit 7 (P7IO7) to "1" to configure the port as an output.

(2) Port 8 mode register (P8IO)

If PWM3OUT is to be used, set bit 7 (P8IO7) to "1" to configure the port as an output. When the PWM function does not operate, this port is fixed at "1".

(3) Port 7 secondary function control register (P7SF)

If PWM1OUT is to be used, set bit 7 (P7SF7) to "1" to configure the port as a secondary function output. In the PWM initial state or when PWM function is halted, the PWM output port is fixed at "1". To fix the port at "0", return the port to a primary function.

(4) Port 8 secondary function control register (P8SF)

If PWM3OUT is to be used, set bit 7 (P8SF7) to "1" to configure the port as a secondary function output. In the PWM initial state or when PWM function is halted, the PWM output port is fixed at "1". To fix the port at "0", return the port to a primary function.

(5) PWM counters (PWC0, PWC1)

Set these counters with the value at which to start counting. Writing to PWC causes the same value to be simultaneously and automatically written to PWCY.

(6) PWM cycle register (PWCY)

Set the PWM cycle in PWCY.

(7) PWM registers (PWR01, PWR23)

If PWM1OUT is to be used, set the desired output duty value in PWR01. If PWM3OUT is to be used, set the desired output duty value in PWR23. Set a value for PWR01 and PWR23 that is larger than the value of PWCY.

(8) PWM control register 0 (PWCON0)

Setting both bits 5 and 6 (PWCK10 and PWCK11) to "1" cascades the two counters (16-bit mode) so that overflow of PWM counter 0 (PWC0) is the clock input to PWM counter 1 (PWC1), thereby forming 16-bit PWM counter (PWC). Bits 1 and 2 (PWCK00 and PWCK01) specify the count clock. Bits 3 and 7 (PWC0OV and PWC1OV) specify the interrupt factor for PWINT0 and PWINT1 interrupt requests. Leaving bit 4 (PW1RUN) set to "1" allows starting and stopping during the 16-bit mode to be controlled with only bit 0 (PW0RUN).

(9) PWM control register 1 (PWCON1)

Bit 0 (PWHSM) specifies normal 16-bit mode or high-speed mode. During the high-speed mode, starting and stopping can be controlled with only bit 4 (PW1RUN) of PWCON0.

[Equation to Calculate 16-Bit PWM Cycle]

$f_{(PWM16)} = PWCLK / (65536 - PWCY)$

$f_{(PWM16)}$: PWM cycle [Hz]
PWCLK : PWM input clock frequency [Hz]
PWCY : Value of PWCY (16 bits)

11.4 PWM Operation

11.4.1 PWM Operation During 8-bit Mode

During the 8-bit mode, PWM output can use the four output pins of PWM0OUT through PWM3OUT.

PWM is started by setting the corresponding RUN bit (PW0RUN, PW1RUN) to "1". When the corresponding RUN bit becomes 1, PWC0 and/or PWC1 begin counting, at the same time the output flip-flop is set to "1", and a High level is output from the PWMnOUT pin (where n = 0 to 3). PWC0 and PWC1 continue to count upward. When their value matches the contents of the corresponding PWRn, an interrupt request is generated, the output flip-flop is reset to "0", and a Low level is output from the PWMnOUT pin. If PWC0 and PWC1 overflow, the output flip-flop is set to "1", and the PWMnOUT pin outputs a High level. Also, the value of PWCY0 and PWCY1 is loaded into PWC0 and PWC1. Thereafter, until the RUN bit is reset to "0", this operation will repeat and the duty controlled waveform will be output from the PWMnOUT pin. When the RUN bit is reset to "0", the PWMnOut pin is fixed at "1".

[Note]

Depending upon the count clock selected for PWC0 and PWC1, immediately after PWM is started, the PWM output duty may be shortened (for one cycle only).
If the value of PWC0 and PWC1 is 00H, and the value of the corresponding PWRn is 00H, the duty output is 1/256. Increasing the value of PWRn increases the output duty (High level). If the value of PWRn is FFH, the output is 256/256 or 100% duty. To realize 0/256 or 0% duty, use the port 1 primary function since 0% duty cannot be realized with the PWM function.

Figure 11-4 shows an example of PWM output operation.

11.4.2 PWM Operation During 16-bit Mode

During the 16-bit mode, PWM output can use the two output pins of PWM1OUT and PWM3OUT.

PWM is started by first setting PW1RUN to "1", and then by setting PWM0RUN to "1". When the RUN bit becomes 1, PWC begins counting, the output flip-flop is simultaneously set to "1", and a High level is output from the PWM1OUT pin (PWM3OUT pin). PWC continues to count upward. When its value matches the contents of PWR01 (PWR23), a PWINT1 (PWINT3) interrupt request is generated, the output flip-flop is reset to "0", and a Low level is output from the PWM1OUT pin (PWM3OUT pin). If PWC overflows, the output flip-flop is set to "1", and the PWM1OUT pin (PWM3OUT pin) outputs a High level. Also, the value of PWCY is loaded into PWC. Thereafter, until the RUN bit is reset to "0", this operation will repeat and the duty controlled waveform will be output from the PWM1OUT pin (PWM3OUT pin). When the RUN bit is reset to "0", the PWMnOut pin is fixed at "1".

However, even in the 16-bit mode, an interrupt request (PWINT2) is generated when the value of PWC0 (lower 8 bits of PWC) matches that of PWR2 (lower 8 bits of PWR23). Also, a PWINT0 interrupt is generated when the value of PWC0 (lower 8 bits of PWC) matches that of PWR0 (lower 8 bits of PWR01), and an interrupt request (PWINT0) is generated when PWC0 overflows.

[Note]

Depending upon the count clock selected for PWC, immediately after PWM is started, the PWM output duty may be shortened (for one cycle only).

If the value of PWC is 0000H, and the value of PWR01 (PWR23) is 0000H, the duty output is 1/65536. Increasing the value of PWR01 (PWR23) increases the output duty (High level). If the value of PWR01 (PWR23) is FFFFH, the output is 65536/65536 or 100% duty. To realize 0/65536 or 0% duty, use the port 1 primary function since 0% duty cannot be realized with the PWM function.

Figure 11-4 shows an example of PWM output operation. Figure 11-5 shows an example of the timing at which PWM output changes.

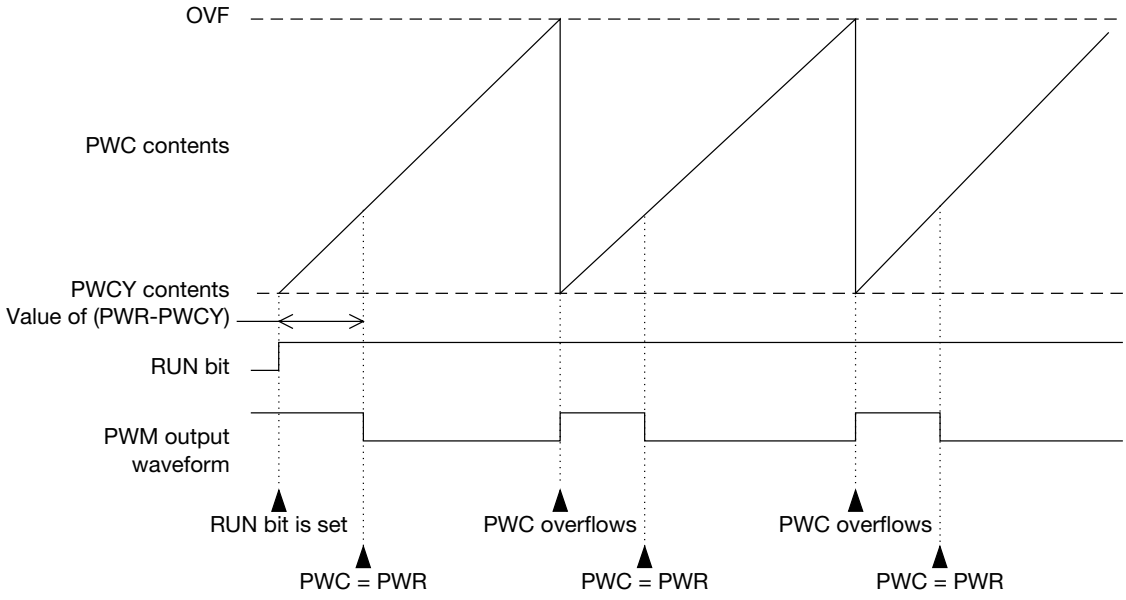


Figure 11-4 Example of PWM Output Operation

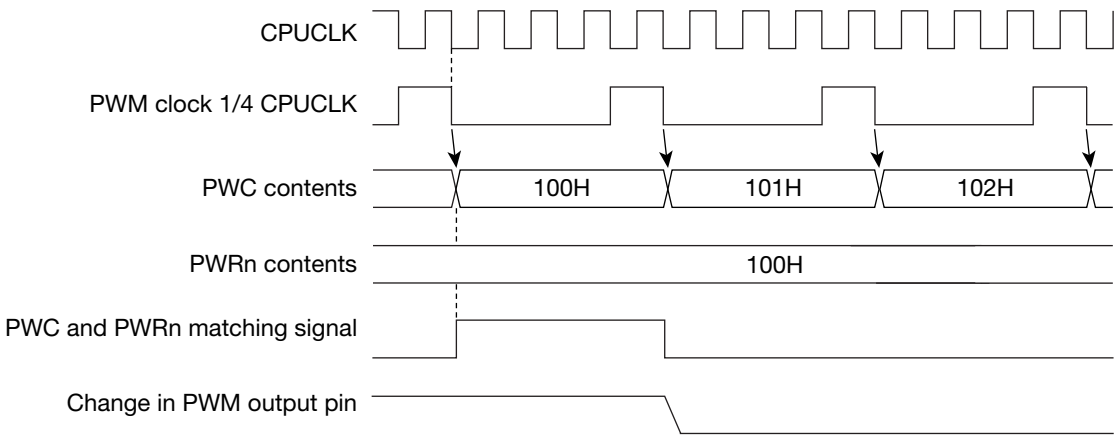


Figure 11-5 Example of PWM Output Change Timing

11.4.3 PWM Operation During High-Speed Mode

During the 16-bit mode, setting bit 0 (PWHSM) of PWCON1 to "1" changes the mode to the high-speed mode. In the high-speed mode, as shown in Figure 11-6, overflow of the upper 8 bits of PWC cause the lower 8 bits of PWC to be incremented. The contents of PWC and PWR are compared, and a High level is output while PWC ≤ PWR.

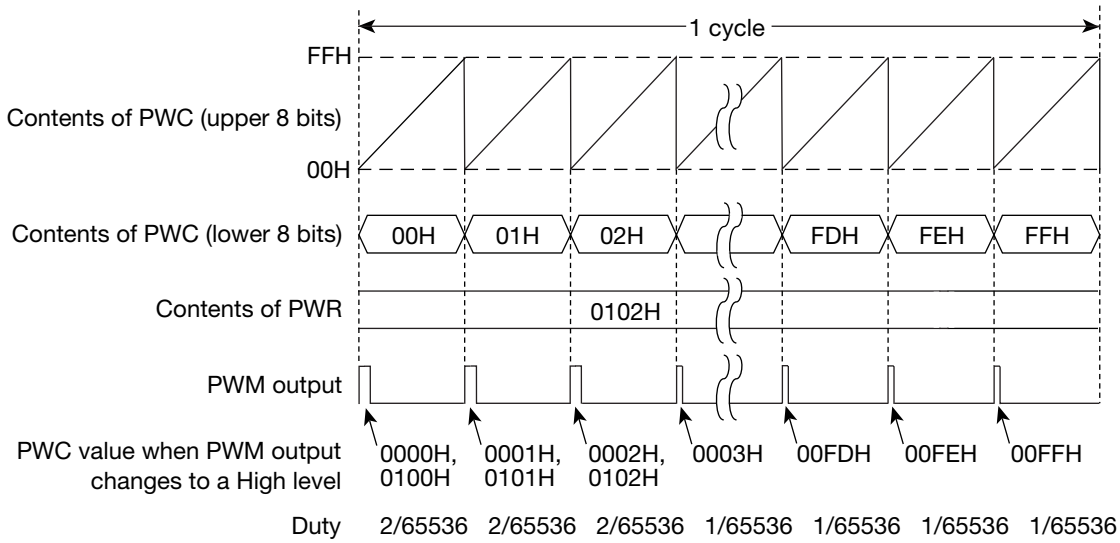


Figure 11-6 PWM Output Waveform During High-Speed Mode

The PWM output in the normal 16-bit mode is 1 pulse per cycle as specified by PWCY. Therefore, when PWCY is 0000H (longest cycle), the PWM output is approximately 458 Hz (for a main clock of 30 MHz). In the high-speed mode, a maximum of 256 pulses are output in the cycle specified by PWCY. The PWM output can achieve the high-speed of 117.2 kHz (for a main clock of 30 MHz). With 256 pulses, because the sum of High and Low intervals is the same as for the 16-bit mode, there is no change in PWM resolution.

Figure 11-7 shows an example of PWM output during the high-speed mode when PWCY is 0000H (longest cycle).

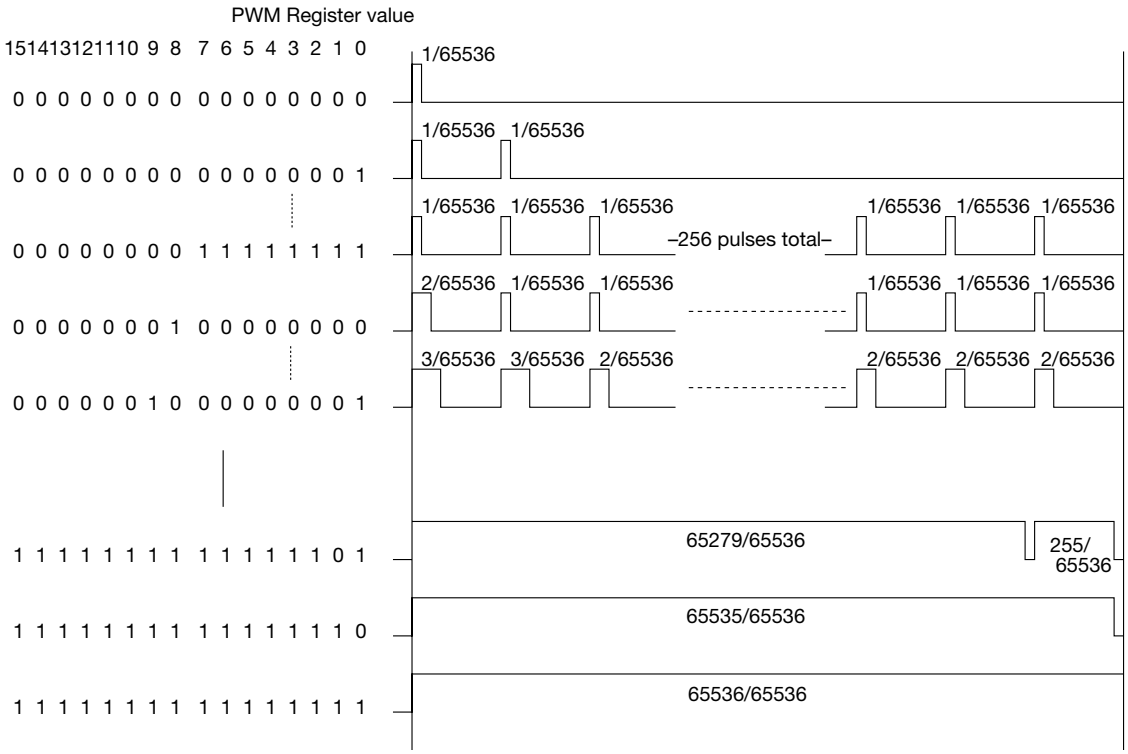


Figure 11-7 Example of PWM Output During High-Speed Mode

11.5 PWM Interrupts

When each PWM interrupt factor occurs, the corresponding interrupt request flag is set to "1". Interrupt request flags are located in interrupt request register 4 (IRQ4).

Interrupts can be enabled or disabled by the interrupt enable flag corresponding to each interrupt factor. The interrupt enable flags are located in interrupt enable register 4 (IE4).

Three levels of priority can be set with the interrupt priority setting flag corresponding to each interrupt factor. The interrupt priority setting flags are located in interrupt priority control register 8 (IP8).

Table 11-2 lists the vector address of each PWM interrupt factor and the interrupt processing flags.

Table 11-2 PWM Vector Addresses and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of PWC0	006A	QPWM0	EPWM0	P1PWM0	P0PWM0
Match of PWC0 and PWR0					
Overflow of PWC1	006C	QPWM1	EPWM1	P1PWM1	P0PWM1
Match of PWC1 and PWR1					
Match of PWC0 and PWR2	006E	QPWM2	EPWM2	P1PWM2	P0PWM2
Match of PWC1 and PWR3	0070	QPWM3	EPWM3	P1PWM3	P0PWM3
Symbols (BYTE) of registers that contain interrupt processing flags		IRQ4	IE4	IP8	
	Reference page	17-16	17-21	17-30	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 12

Serial Port Functions

12. Serial Port Functions

12.1 Overview

The MSM66577 family contains four built-in serial port channels: two UART/Synchronous receiver transmitter serial port channels (SIO1 and SIO6), and two channels (SIO4 and SIO5) of synchronous receiver transmitter serial ports with 32-byte FIFO.

12.2 Serial Port Configuration

Figure 12-1 shows the configuration of the serial ports.

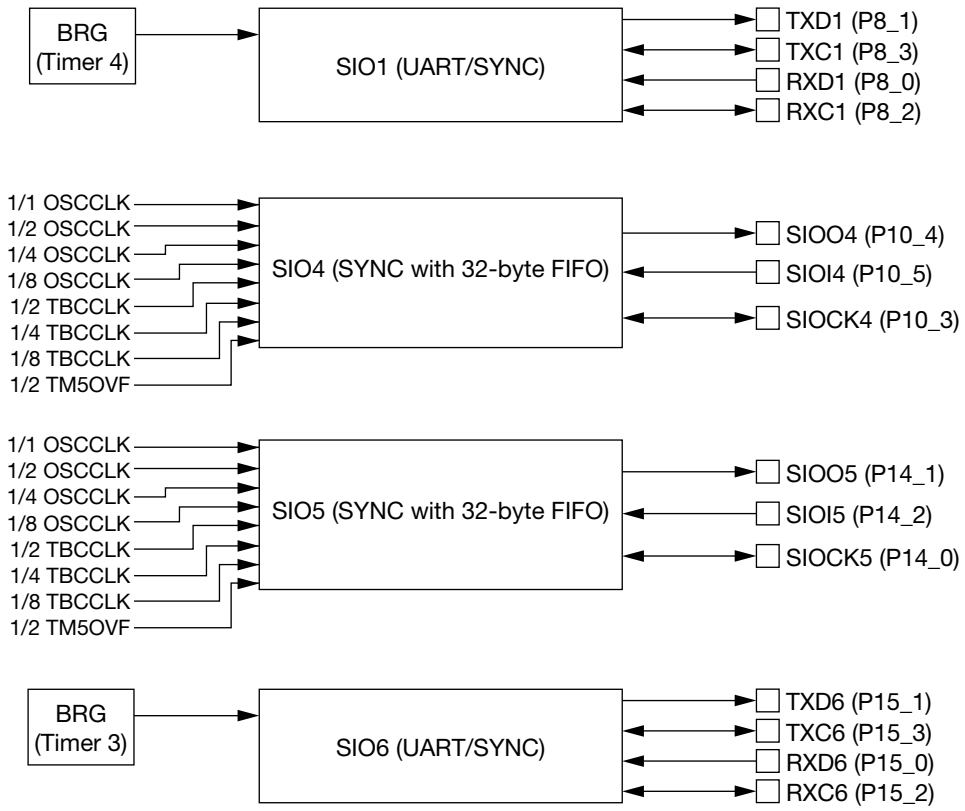


Figure 12-1 Serial Port Configuration

12.3 Serial Port Registers

Table 12-1 lists a summary of SFRs for control of the serial port functions.

Table 12-1 Summary of SFRs for Serial Port Function Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0084☆	SIO1 transmit control register	ST1CON	—	R/W	8	04	12-4
0085	SIO1 receive control register	SR1CON	—	R/W	8	00	12-6
0086	SIO1 transmit-receive buffer register	S1BUF	—	R/W	8	Undefined	12-10
0087☆	SIO1 status register	S1STAT	—	R/W	8	00	12-8
008C☆	SIO4 control register	SIO4CON	—	R/W	8	80	12-41
008D	FIFO control register	FIFOCON	—	R/W	8	11	12-43
008E	SIO4 serial input FIFO data register	SIN4	—	R	8	Undefined	12-45
008F	SIO4 serial output FIFO data register	SOUT4	—	W	8	Undefined	12-45
00CF☆	FIFO mode control register	FIFOMOD	—	R/W	8	FC	12-54
00D5☆	SIO5 control register	SIO5CON	—	R/W	8	80	12-52
00D6	SIO5 serial input FIFO data register	SIN5	—	R	8	Undefined	12-54
00D7	SIO5 serial output FIFO data register	SOUT5	—	W	8	Undefined	12-54
00F4☆	SIO6 transmit control register	ST6CON	—	R/W	8	04	12-16
00F5	SIO6 receive control register	SR6CON	—	R/W	8	00	12-18
00F6	SIO6 transmit-receive buffer register	S6BUF	—	R/W	8	Undefined	12-22
00F7☆	SIO6 status register	S6STAT	—	R/W	8	00	12-20

[Notes]

1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

12.4 SIO1

The SIO1 has a UART mode and a synchronous mode. Timer 4 is used as a baud rate generator exclusively for SIO1.

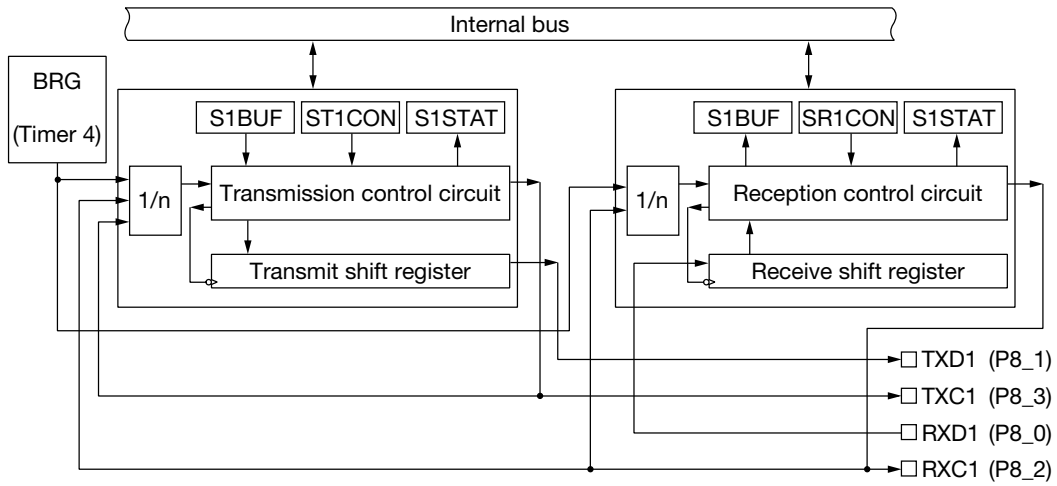
Table 12-2 lists specifications of SIO1.

Table 12-2 SIO1 Specifications

	UART mode	Synchronous mode
Data length	Selectable as 7 or 8 bits	Selectable as 7 or 8 bits
Parity	Odd, even, none	
Error service	Parity, overrun, framing	Overrun
Stop bit	Selectable as 1 or 2 bits	
Factors that generate interrupt requests	Transmit buffer empty, transmit complete, receive complete	Transmit buffer empty, transmit complete, receive complete
Full-duplex communication	Possible	Possible
Transmit-receive buffer	Both transmission and reception data are double buffered	Both transmission and reception data are double buffered
Max. communication speed (f = 30MHz)	1.875 Mbps	7.5 Mbps
Other	LSB first An external clock can be used for the UART baud rate	LSB first Master mode/ slave mode

12.4.1 SIO1 Configuration

Figure 12-2 shows the SIO1 configuration.



BRG: Baud rate generator (timer 4)
S1BUF: Transmit-receive buffer register
ST1CON: SIO1 transmit control register
SR1CON: SIO1 receive control register
S1STAT: SIO1 status register
1/n: 1/n frequency dividing counter
TXD1: SIO1 transmit data output pin (P8_1)
TXC1: SIO1 transmit clock I/O pin (P8_3)
RXD1: SIO1 receive data input pin (P8_0)
RXC1: SIO1 receive clock I/O pin (P8_2)

Figure 12-2 SIO1 Configuration

12.4.2 Description of SIO1 Registers

(1) SIO1 transmit control register (ST1CON)

The SIO1 transmit control register (ST1CON) is a 7-bit register that controls operation of SIO1 transmission.

ST1CON can be read from and written to by the program. However, write operations are invalid for bit 2. If read, a value of "1" will always be obtained for bit 2.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ST1CON becomes 04H, the data length for SIO1 transmission is 8-bits, 2 stop bits are selected and the mode changes to UART mode with no parity.

The baud rate source is the same for transmission and reception. It is set by the receive control register (SR1CON) to be described later.

[Note]
If ST1CON is to be modified, make those changes after transmission is complete. If ST1CON is modified before transmission is completed, the current transmission and future transmissions will not be executed correctly.

Figure 12-3 shows the ST1CON configuration.

[Description of each bit]
- ST1MOD (bit 0)
  ST1MOD specifies the transmission mode (UART or synchronous).
- ST1LN (bit 1)
  ST1LN specifies the SIO1 transmit data length.
- ST1STB/ST1SLV (bit 3)
  During the UART mode, ST1STB specifies the SIO1 stop bit length.
  During the synchronous mode, ST1SLV specifies master or slave operation.
- ST1PEN (bit 4)
  ST1PEN specifies whether there is parity during SIO1 transmission. (Only valid during the UART mode)
- ST1ODD (bit 5)
  ST1ODD specifies the parity bit logic during SIO1 transmission. (Only valid during the UART mode)
- TR1MIE (bit 6)
  TR1MIE specifies whether to use the SIO1 transmit buffer empty signal as an interrupt request signal.
- TR1NIE (bit 7)
  TR1NIE specifies whether to use the SIO1 transmit complete signal as an interrupt request signal.

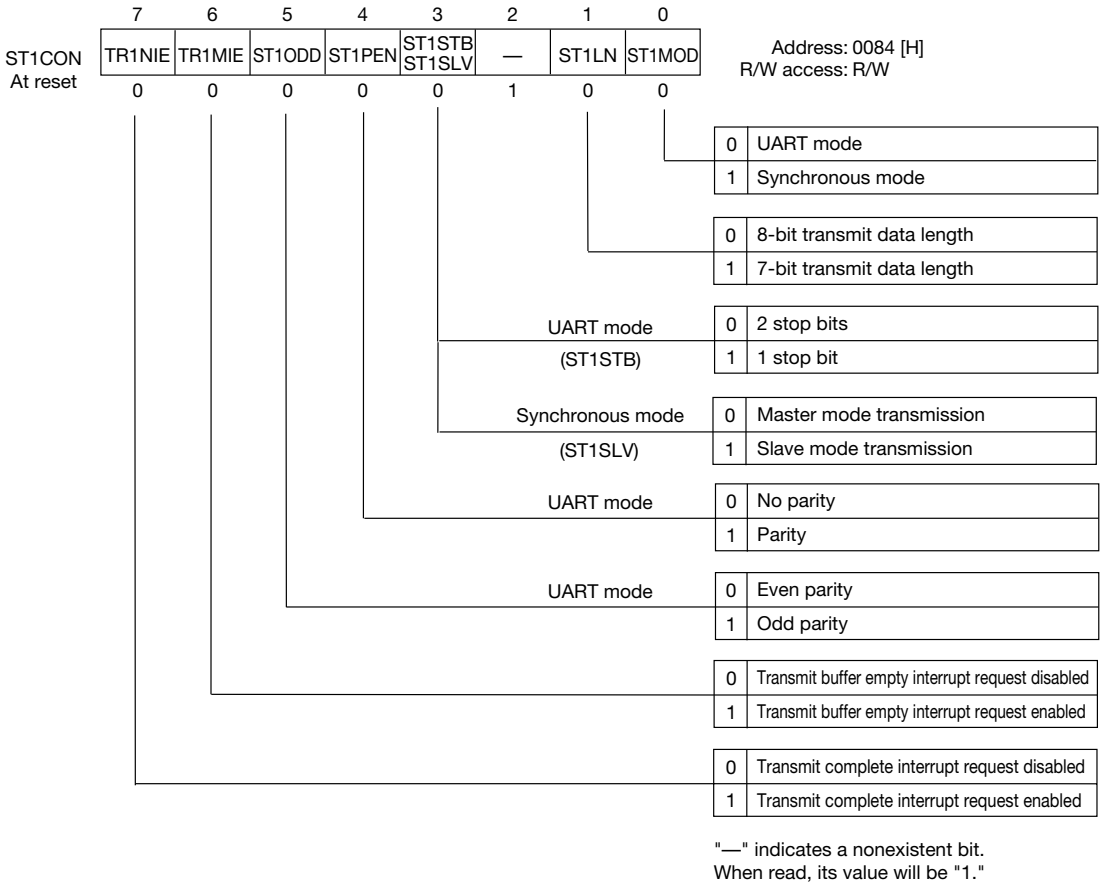


Figure 12-3 ST1CON Configuration

(2) SIO1 receive control register (SR1CON)

The SIO1 receive control register (SR1CON) is an 8-bit register that controls operation of SIO1 reception.

SR1CON can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SR1CON becomes 00H and SIO1 reception is disabled.

[Note]
If SR1CON is to be modified, first reset SR1REN (bit 7) to "0" and then implement the change. If SR1CON is modified before SR1REN (bit 7) is reset to "0", the current reception and future receptions will not be executed correctly.

Figure 12-4 shows the SR1CON configuration.

[Description of each bit]

- SR1MOD (bit 0)
  SR1MOD specifies the SIO1 reception mode (UART or synchronous).
- SR1LN (bit 1)
  SR1LN specifies the SIO1 receive data length.
- S1EXC (bit 2)
  S1EXC specifies the baud rate clock to be used by SIO1 during the UART mode. (This clock is the same for both transmission and reception. The shift clock has a frequency 1/16th of the clock specified here.)
- SR1SLV (bit 3)
  During the synchronous mode, ST1SLV specifies master or slave operation of SIO1. (Only valid during the synchronous mode)
- SR1PEN (bit 4)
  SR1PEN specifies whether there is parity during SIO1 reception. (Only valid during the UART mode)
- SR1ODD (bit 5)
  SR1ODD specifies the parity bit logic during SIO1 reception. (Only valid during the UART mode)
- RC1IE (bit 6)
  RC1IE specifies whether to use the SIO1 receive complete signal as an interrupt request signal.
- SR1REN (bit 7)
  SR1REN enables or disables SIO1 reception.

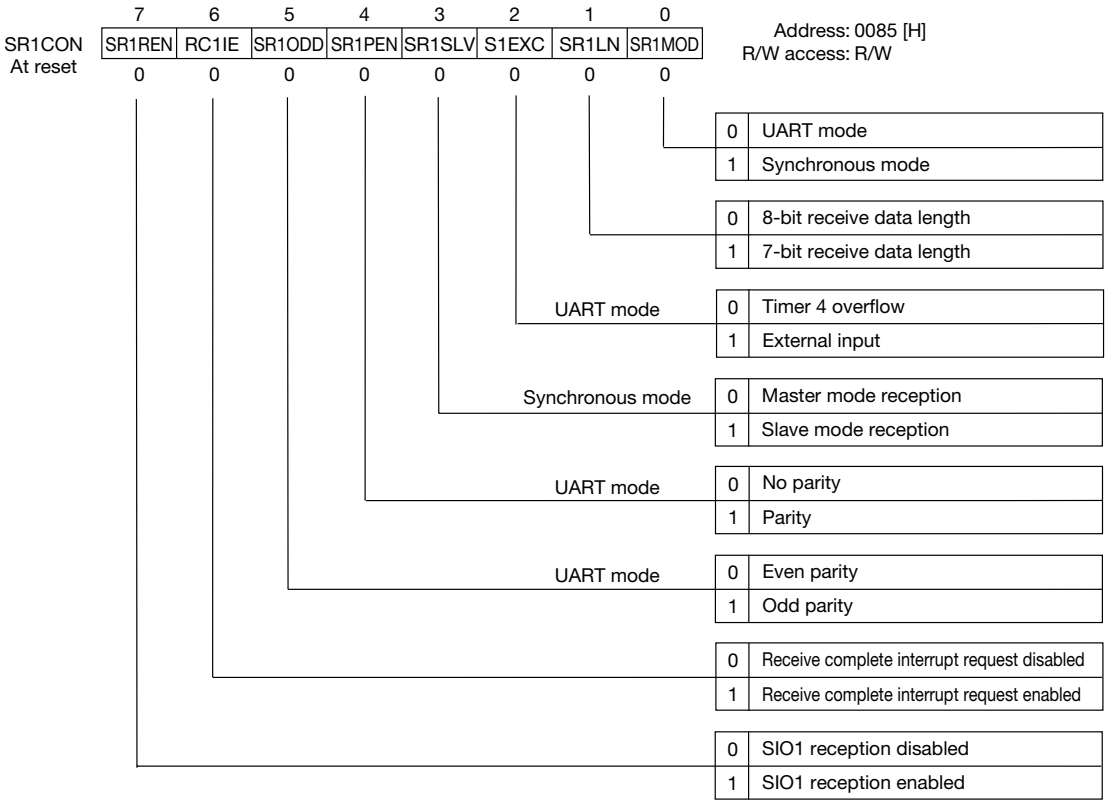


Figure 12-4 SR1CON Configuration

(3) SIO1 status register (S1STAT)

The SIO1 status register (S1STAT) consists of 6 bits. Bits 0 through 2 save the SIO1 status (normal or error) after reception is completed. Bits 3 through 5 save the status of SIO1 at the start and completion of transmission and reception. However bits 0 through 2 are updated after the reception is completed.

S1STAT can be read from and written to by the program. However, write operations are invalid for bits 6 and 7. If read, a value of "0" will always be obtained for bits 6 and 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), S1STAT becomes 00H.

Figure 12-5 shows the S1STAT configuration.

[Description of each bit]

- FERR1 (bit 0)
  If the stop bit in the data received by SIO1 is "0", FERR1 is set to "1" (framing error). This bit is only valid during the UART mode.
- OERR1 (bit 1)
  When the SIO1 reception is complete, if the previously received data has not been read by the program, OERR1 is set to "1" (overrun error).
- PERR1 (bit 2)
  If the parity bit in the data received by SIO1 does not match the parity of the data, PERR1 is set to "1" (parity error). This bit is only valid during the UART mode.
- TR1EMP (bit 3)
  If the SIO1 transmit buffer empty signal is generated, TR1EMP is set to "1".
- TR1END (bit 4)
  If the SIO1 transmit complete signal is generated, TR1EMP is set to "1".
- RC1END (bit 5)
  If the SIO1 receive complete signal is generated, RC1END is set to "1".

[Note]

Once each bit of S1STAT is set to "1", the hardware does not reset the bits to "0". Therefore, reset the bits to "0" with the program.

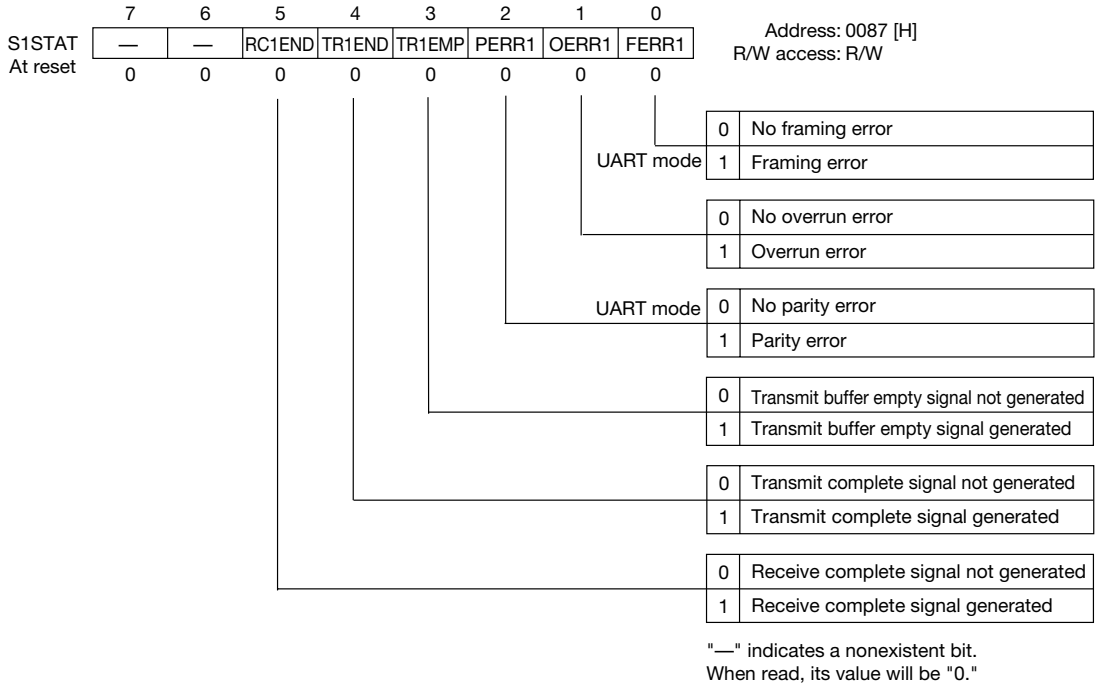


Figure 12-5 S1STAT Configuration

(4) SIO1 transmit-receive buffer register (S1BUF)

The SIO1 transmit-receive buffer register (S1BUF) is an 8-bit register that stores the transmit and receive data for serial port transmission and reception. Because S1BUF has a duplex configuration for transmission and reception, it operates as a transmission buffer when written to, and as a reception buffer when read from.

After the transmit data has been written to S1BUF, the transmit data is transferred to the transmit shift register and the transmit buffer empty signal is generated. At that time, SIO1 will begin transmission.

After reception is complete, the contents of the receive shift register are transferred to S1BUF and at that time, the receive complete signal is generated. The contents of S1BUF are saved until the next reception is completed.

During a 7-bit data reception, bit 7 of S1BUF is "1", and the 7 bits from bit 0 through bit 6 are the reception data.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of S1BUF is undefined.

(5) SIO1 transmit shift register, receive shift register

The transmit shift register and receive shift register are 8-bit shift registers that perform the actual shifting operation during transmission and reception.

The transmit shift register and receive shift register cannot be read from or written to by the program.

Table 12-3 lists SIO1 transmit-receive frame lengths.

Table 12-3 SIO1 Transmit-Receive Frame Lengths

ST1CON/SR1CON				Transmit/Receive Frame Length
ST1PEN SR1PEN	ST1STB	ST1LN SR1LN	ST1MOD SR1MOD	1 2 3 4 5 6 7 8 9 10 11 12 [bit]
0	0	0	0	START 8-bit data STOP STOP
0	0	1	0	START 7-bit data STOP STOP
0	1	0	0	START 8-bit data STOP
0	1	1	0	START 7-bit data STOP
1	0	0	0	START 8-bit data PARITY STOP STOP
1	0	1	0	START 7-bit data PARITY STOP STOP
1	1	0	0	START 8-bit data PARITY STOP
1	1	1	0	START 7-bit data PARITY STOP
—	—	0	1	8-bit data
—	—	1	1	7-bit data

12.4.3 Example of SIO1-related Register Settings

12.4.3.1 UART Mode Settings

- **Transmit settings**

(1) Port 8 mode register (P8IO)
If TXD8 (transmit data output) is to be used, set bit 1 (P8IO1) to "1" to configure that port as an output. If the baud rate clock is to be input externally, reset bit 2 (P8IO2) to "0" to configure that port as an input.

(2) Port 8 secondary function control register (P8SF)
If TXD1 (transmit data output) is to be used, set bit 1 (P8SF1) to "1" to configure that port as a secondary function output. If the baud rate clock is to be input externally, specify with bit 2 (P8SF2) whether the input will be pulled-up.

(3) SIO1 transmit control register (ST1CON)
Reset bit 0 (ST1MOD) to "0" to change the mode to UART mode. Specify the transmit data length with bit 1 (ST1LN). Specify the stop bit length with bit 3 (ST1STB). Specify whether there is parity with bit 4 (ST1PEN). If parity is selected, specify the parity bit logic with bit 5 (ST1ODD). With bit 6 (TR1MIE), specify whether interrupt requests are enabled or disabled when a transmit buffer empty signal occurs. With bit 7 (TR1NIE), specify whether interrupt requests are enabled or disabled when a transmit complete signal occurs.

(4) SIO1 receive control register (SR1CON)
Specify with bit 2 (S1EXC) whether the baud rate clock is internal (overflow output of timer 4) or external (RXC1).

(5) SIO1 transmit-receive buffer register (S1BUF)
Transmission is started by writing the transmit data to S1BUF.

- **Receive settings**

(1) Port 8 mode register (P8IO)
If RXD1 (receive data input) is to be used, reset bit 0 (P8IO0) to "0" to configure that port as an input. If the baud rate clock is to be input externally, reset bit 2 (P8IO2) to "0" to configure that port as an input.

(2) Port 8 secondary function control register (P8SF)
Specify with bit 1 (P8SF0) whether the RXD1 pin will be pulled-up. If the baud rate clock is to be input externally, specify with bit 2 (P8SF2) whether the input will be pulled-up.

(3) SIO1 receive control register (SR1CON)

Reset bit 0 (SR1MOD) to "0" to change the mode to UART mode. Specify the receive data length with bit 1 (SR1LN). Specify with bit 2 (S1EXC) whether the baud rate clock is internal (overflow output of timer 4) or external (RXC1). Specify whether there is parity with bit 4 (SR1PEN). If parity is selected, specify the parity bit logic with bit 5 (SR1ODD). With bit 7 (RC1IE), specify whether interrupt requests are enabled or disabled when a receive complete signal occurs. If bit 7 (SR1REN) is set to "1", reception is enabled and the reception operation is performed when data arrives.

12.4.3.2 Synchronous Mode Settings

- **Transmit settings**

(1) Port 8 mode register (P8IO)

If TXD1 (transmit data output) is to be used, set bit 1 (P8IO1) to "1" to configure that port as an output. If the transmit clock is to be output externally (master mode), set bit 3 (P8IO3) to "1" to configure that port as an output. If the baud rate clock is to be input externally (slave mode), reset bit 3 (P8IO3) to "0" to configure that port as an input.

(2) Port 8 secondary function control register (P8SF)

If TXD1 (transmit data output) is to be used, set bit 1 (P8SF1) to "1" to configure that port as a secondary function output. If the transmit clock is to be output externally (master mode), set bit 3 (P8SF3) to "1" to configure that port as a secondary function output. If the baud rate clock is to be input externally (slave mode), specify with bit 3 (P8SF3) whether the input will be pulled-up.

(3) SIO1 transmit control register (ST1CON)

Set bit 0 (ST1MOD) to "1" to specify the mode to synchronous mode. Specify the transmit data length with bit 1 (ST1LN). Specify master or slave mode transmission with bit 3 (ST1STB). With bit 6 (TR1MIE), specify whether interrupt requests are enabled or disabled when a transmit buffer empty signal occurs. With bit 7 (TR1NIE), specify whether interrupt requests are enabled or disabled when a transmit complete signal occurs.

(4) SIO1 transmit-receive buffer register (S1BUF)

Transmission is started by writing the transmit data to S1BUF.

- **Receive settings**

(1) Port 8 mode register (P8IO)

If RXD1 (receive data input) is to be used, reset bit 0 (P8IO0) to "0" to configure that port as an input. If the transmit clock is to be output externally (master mode), set bit 2 (P8IO2) to "1" to configure that port as an output. If the transmit clock is to be input externally (slave mode), reset bit 2 (P8IO2) to "0" to configure that port as an input.

(2) Port 8 secondary function control register (P8SF)

Specify with bit 0 (P8SF0) whether the RXD1 pin will be pulled-up. If the transmit clock is to be output externally (master mode), set bit 2 (P8SF2) to "1" to configure that port as a secondary function output. If the transmit clock is to be input externally (slave mode), specify with bit 2 (P8SF2) whether the input will be pulled-up.

(3) SIO1 receive control register (SR1CON)

Set bit 0 (SR1MOD) to "1" to specify the mode to synchronous mode. Specify the receive data length with bit 1 (SR1LN). Specify the master or slave mode with bit 3 (SR1SLV). With bit 6 (RC1IE), specify whether interrupt requests are enabled or disabled when a receive complete signal occurs. If bit 7 (SR1REN) is set to "1", reception is enabled and the reception operation is performed when data arrives.

12.4.3.3 Baud Rate Generator (Timer 4) Settings

If overflow of timer 4 is selected for use as the baud rate clock, implement the following settings.

(1) General-purpose 8-bit timer 4 counter (TM4C)

Set the timer value that will be valid at the start of counting. When writing to TM4C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 4 register (TM4R).

(2) General-purpose 8-bit timer 4 control register (TM4CON)

Bits 0 to 2 (TM4C0 to TM4C2) of this register specify the count clock for timer 4. If bit 3 (TM4RUN) is set to "1", timer 4 will begin counting. If reset to "0", timer 4 will halt counting.

[Equation to Calculate Baud Rate]

$B = f_{(TM4)} \times 1/(256 - D) \times 1/n$

B : baud rate [bps]
$f_{(TM4)}$: timer 4 input clock frequency [Hz]
D : reload value (0 to 255)
n : 16 for the UART mode
4 for the synchronous mode

12.4.4 SIO1 Interrupt

When any SIO1 interrupt factor occurs, the interrupt request flag (QSIO1) is set to "1". The interrupt request flag (QSIO1) is located in interrupt request register 2 (IRQ2).

Interrupts can be enabled or disabled by the interrupt enable flag (ESIO1). The interrupt enable flag (ESIO1) is located in interrupt enable register 2 (IE2).

Three levels of priority can be set with the interrupt priority setting flags (P0SIO1 and P1SIO1). The interrupt priority setting flags (P0SIO1 and P1SIO1) are located in interrupt priority control register 5 (IP5).

Table 12-4 lists the vector address of the SIO1 interrupt factors and the interrupt processing flags.

Table 12-4 SIO1 Vector Address and Interrupt Processing Flags

<table>
<tr><td rowspan="2">Interrupt factor</td><td rowspan="2">Vector address [H]</td><td rowspan="2">Interrupt request</td><td rowspan="2">Interrupt enable</td><td colspan="2">Priority level</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td>SIO1 transmit buffer empty signal is generated</td><td rowspan="3">0038</td><td rowspan="3">QSIO1</td><td rowspan="3">ESIO1</td><td rowspan="3">P1SIO1</td><td rowspan="3">P0SIO1</td></tr>
<tr><td>SIO1 transmit complete signal is generated</td></tr>
<tr><td>SIO1 receive complete signal is generated</td></tr>
<tr><td colspan="2" rowspan="2">Symbols (byte) of registers that contain interrupt processing flags</td><td>IRQ2</td><td>IE2</td><td colspan="2">IP5</td></tr>
<tr><td>Reference page</td><td>17-14</td><td>17-19</td><td colspan="2">17-27</td></tr>
</table>

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

12.5 SIO6

The SIO6 has a UART mode and a synchronous mode. Timer 3 is used as a baud rate generator exclusively for SIO6.

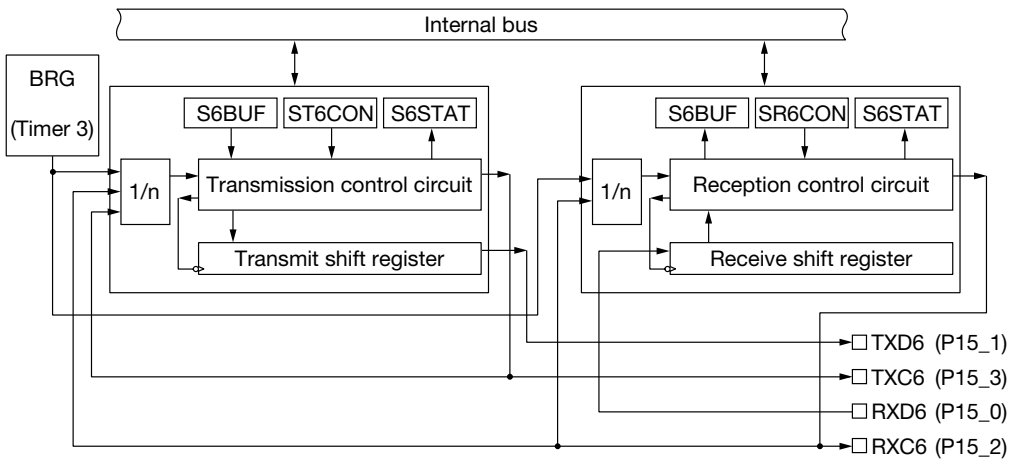
Table 12-5 lists specifications of SIO6.

Table 12-5 SIO6 Specifications

	UART mode	Synchronous mode
Data length	Selectable as 7 or 8 bits	Selectable as 7 or 8 bits
Parity	Odd, even, none	
Error service	Parity, overrun, framing	Overrun
Stop bit	Selectable as 1 or 2 bits	
Factors that generate interrupt requests	Transmit buffer empty, transmit complete, receive complete	Transmit buffer empty, transmit complete, receive complete
Full-duplex communication	Possible	Possible
Transmit-receive buffer	Both transmission and reception data are double buffered	Both transmission and reception data are double buffered
Max. communication speed (f = 30 MHz)	1.875 Mbps	7.5 Mbps
Other	LSB first An external clock can be used for the UART baud rate	LSB first Master mode/ slave mode

12.5.1 SIO6 Configuration

Figure 12-6 shows the SIO6 configuration.



BRG: Baud rate generator (timer 3)
S6BUF: Transmit-receive buffer register
ST6CON: SIO6 transmit control register
SR6CON: SIO6 receive control register
S6STAT: SIO6 status register
1/n: 1/n frequency dividing counter
TXD6: SIO6 transmit data output pin (P15_1)
TXC6: SIO6 transmit clock I/O pin (P15_3)
RXD6: SIO6 receive data input pin (P15_0)
RXC6: SIO6 receive clock I/O pin (P15_2)

Figure 12-6 SIO6 Configuration

12.5.2 Description of SIO6 Registers

(1) SIO6 transmit control register (ST6CON)

The SIO6 transmit control register (ST6CON) is a 7-bit register that controls operation of SIO6 transmission.

ST6CON can be read from and written to by the program. However, write operations are invalid for bit 2. If read, a value of "1" will always be obtained for bit 2.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ST6CON becomes 04H, the data length for SIO6 transmission is 8-bits, 2 stop bits are selected and the mode changes to UART mode with no parity.

The baud rate source is the same for transmission and reception. It is set by the receive control register (SR6CON) to be described later.

[Note]
If ST6CON is to be modified, make those changes after transmission is complete. If ST6CON is modified before transmission is completed, the current transmission and future transmissions will not be executed correctly.

Figure 12-7 shows the ST6CON configuration.

[Description of each bit]
- ST6MOD (bit 0)
  ST6MOD specifies the transmission mode (UART or synchronous).
- ST6LN (bit 1)
  ST6LN specifies the SIO6 transmit data length.
- ST6STB/ST6SLV (bit 3)
  During the UART mode, ST6STB specifies the SIO6 stop bit length.
  During the synchronous mode, ST6SLV specifies master or slave operation.
- ST6PEN (bit 4)
  ST6PEN specifies whether there is parity during SIO6 transmission. (Only valid during the UART mode)
- ST6ODD (bit 5)
  ST6ODD specifies the parity bit logic during SIO6 transmission. (Only valid during the UART mode)
- TR6MIE (bit 6)
  TR6MIE specifies whether to use the SIO6 transmit buffer empty signal as an interrupt request signal.
- TR6NIE (bit 7)
  TR6NIE specifies whether to use the SIO6 transmit complete signal as an interrupt request signal.

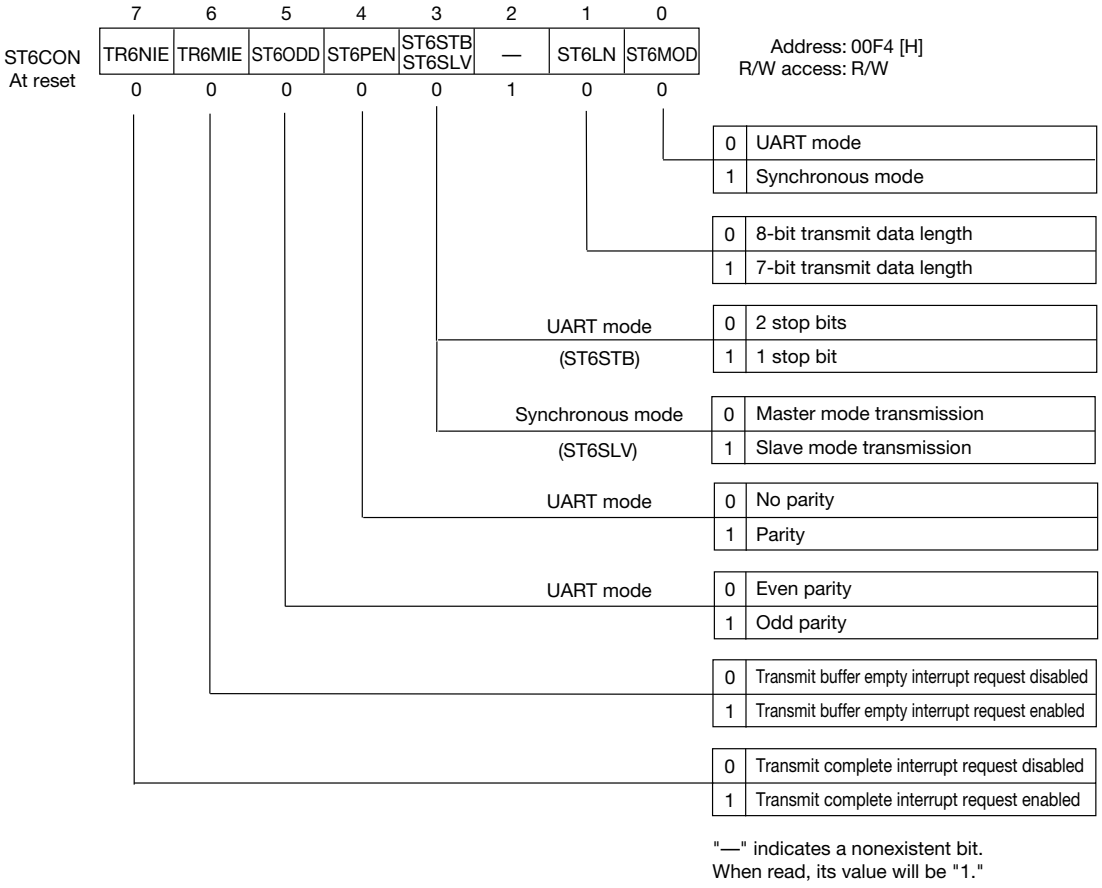


Figure 12-7 ST6CON Configuration

(2) SIO6 receive control register (SR6CON)

The SIO6 receive control register (SR6CON) is an 8-bit register that controls operation of SIO6 reception.

SR6CON can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SR6CON becomes 00H and SIO6 reception is disabled.

[Note]
If SR6CON is to be modified, first reset SR6REN (bit 7) to "0" and then implement the change. If SR6CON is modified before SR6REN (bit 7) is reset to "0", the current reception and future receptions will not be executed correctly.

Figure 12-8 shows the SR6CON configuration.

[Description of each bit]

- SR6MOD (bit 0)
  SR6MOD specifies the SIO6 reception mode (UART or synchronous).
- SR6LN (bit 1)
  SR6LN specifies the SIO6 receive data length.
- S6EXC (bit 2)
  S6EXC specifies the baud rate clock to be used by SIO6 during the UART mode. (This clock is the same for both transmission and reception. The shift clock has a frequency 1/16th of the clock specified here.)
- SR6SLV (bit 3)
  During the synchronous mode, ST6SLV specifies master or slave operation of SIO6. (Only valid during the synchronous mode)
- SR6PEN (bit 4)
  SR6PEN specifies whether there is parity during SIO6 reception. (Only valid during the UART mode)
- SR6ODD (bit 5)
  SR6ODD specifies the parity bit logic during SIO6 reception. (Only valid during the UART mode)
- RC6IE (bit 6)
  RC6IE specifies whether to use the SIO6 receive complete signal as an interrupt request signal.
- SR6REN (bit 7)
  SR6REN enables or disables SIO6 reception.

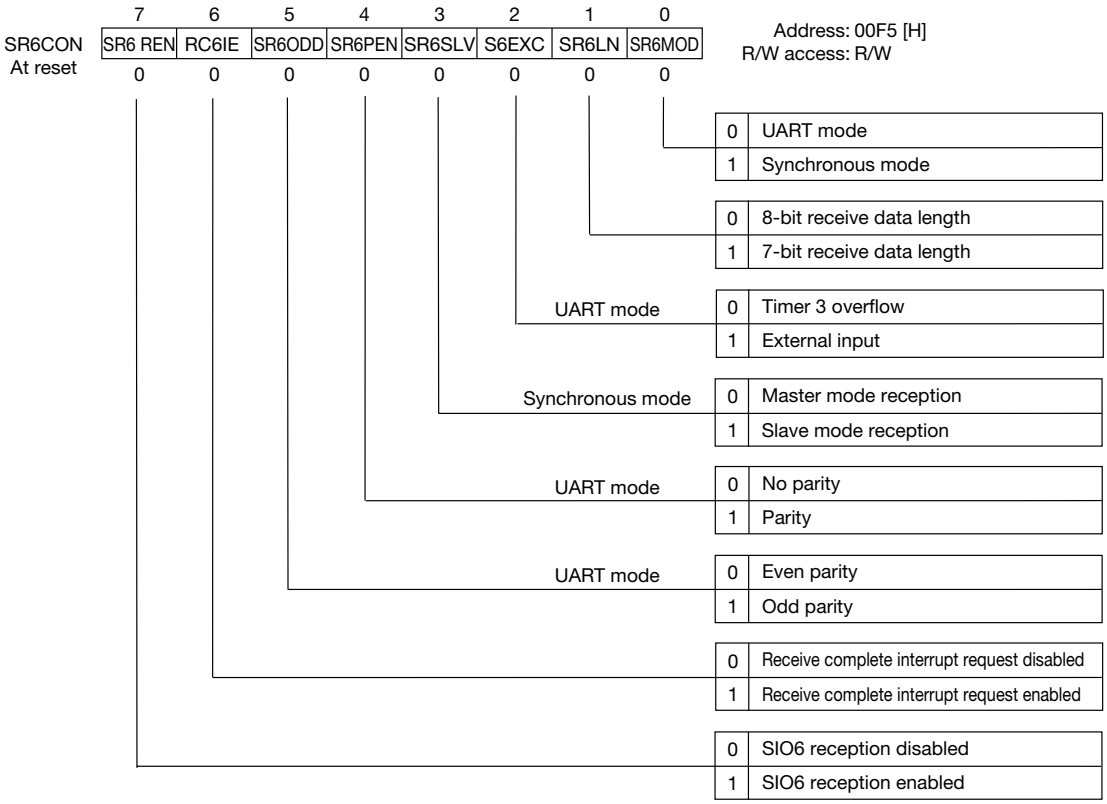


Figure 12-8 SR6CON Configuration

(3) SIO6 status register (S6STAT)

The SIO6 status register (S6STAT) consists of 6 bits. Bits 0 through 2 save the SIO6 status (normal or error) after reception is completed. Bits 3 through 5 save the status of SIO6 at the start and completion of transmission and reception. However bits 0 through 2 are updated after the reception is completed.

S6STAT can be read from and written to by the program. However, write operations are invalid for bits 6 and 7. If read, a value of "0" will always be obtained for bits 6 and 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), S6STAT becomes 00H.

Figure 12-9 shows the S6STAT configuration.

[Description of each bit]

- FERR6 (bit 0)
  If the stop bit in the data received by SIO6 is "0", FERR6 is set to "1" (framing error). This bit is only valid during the UART mode.
- OERR6 (bit 1)
  When the SIO6 reception is complete, if the previously received data has not been read by the program, OERR6 is set to "1" (overrun error).
- PERR6 (bit 2)
  If the parity bit in the data received by SIO6 does not match the parity of the data, PERR6 is set to "1" (parity error). This bit is only valid during the UART mode.
- TR6EMP (bit 3)
  If the SIO6 transmit buffer empty signal is generated, TR6EMP is set to "1".
- TR6END (bit 4)
  If the SIO6 transmit complete signal is generated, TR6EMD is set to "1".
- RC6END (bit 5)
  If the SIO6 receive complete signal is generated, RC6END is set to "1".

[Note]

Once each bit of S6STAT is set to "1", the hardware does not reset the bits to "0". Therefore, reset the bits to "0" with the program.

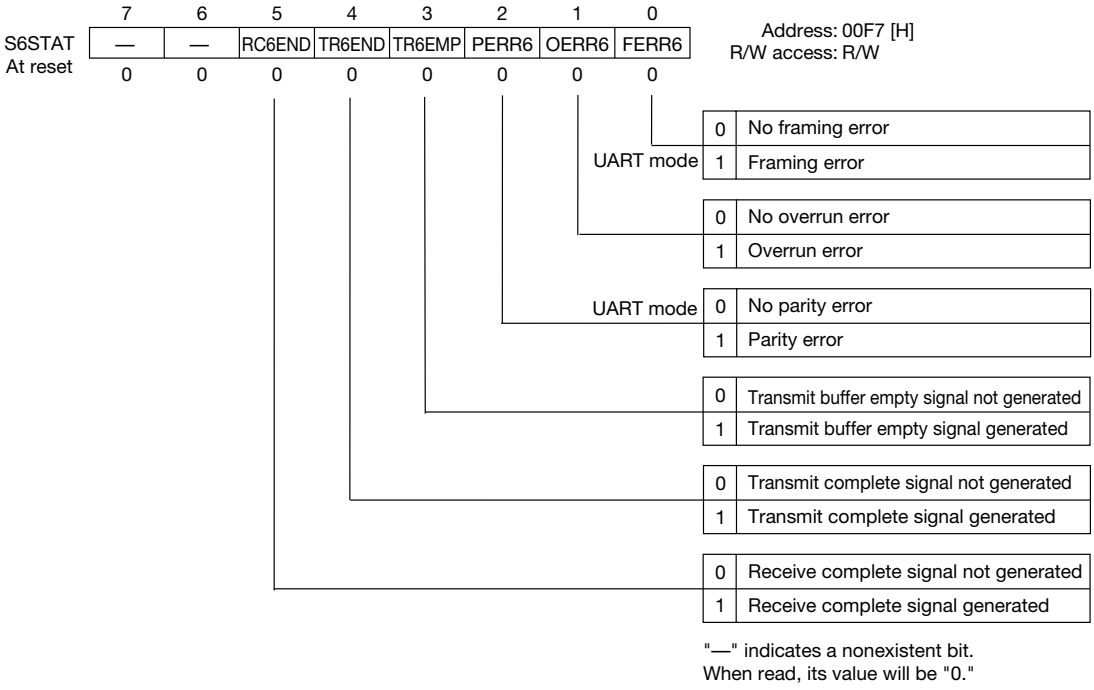


Figure 12-9 S6STAT Configuration

(4) SIO6 transmit-receive buffer register (S6BUF)

The SIO6 transmit-receive buffer register (S6BUF) is an 8-bit register that stores the transmit and receive data for serial port transmission and reception. Because S6BUF has a duplex configuration for transmission and reception, it operates as a transmission buffer when written to, and as a reception buffer when read from.

After the transmit data has been written to S6BUF, the transmit data is transferred to the transmit shift register and the transmit buffer empty signal is generated. At that time, SIO6 will begin transmission.

After reception is complete, the contents of the receive shift register are transferred to S6BUF and at that time, the receive complete signal is generated. The contents of S6BUF are saved until the next reception is completed.

During a 7-bit data reception, bit 7 of S6BUF is "1", and the 7 bits from bit 0 through bit 6 are the reception data.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of S6BUF is undefined.

(5) SIO6 transmit shift register, receive shift register

The transmit shift register and receive shift register are 8-bit shift registers that perform the actual shifting operation during transmission and reception.

The transmit shift register and receive shift register cannot be read from or written to by the program.

Table 12-6 lists SIO6 transmit-receive frame lengths.

Table 12-6 SIO6 Transmit-Receive Frame Lengths

ST6CON/SR6CON				Transmit/Receive Frame Length
ST6PEN SR6PEN	ST6STB	ST6LN SR6LN	ST6MOD SR6MOD	1 2 3 4 5 6 7 8 9 10 11 12 [bit]
0	0	0	0	START 8-bit data STOP STOP
0	0	1	0	START 7-bit data STOP STOP
0	1	0	0	START 8-bit data STOP
0	1	1	0	START 7-bit data STOP
1	0	0	0	START 8-bit data PARITY STOP STOP
1	0	1	0	START 7-bit data PARITY STOP STOP
1	1	0	0	START 8-bit data PARITY STOP
1	1	1	0	START 7-bit data PARITY STOP
—	—	0	1	8-bit data
—	—	1	1	7-bit data

12.5.3 Example of SIO6-related Register Settings

12.5.3.1 UART Mode Settings

- **Transmit settings**

(1) Port 15 mode register (P15IO)
If TXD6 (transmit data output) is to be used, set bit 1 (P15IO1) to "1" to configure that port as an output. If the baud rate clock is to be input externally, reset bit 2 (P15IO2) to "0" to configure that port as an input.

(2) Port 15 secondary function control register (P15SF)
If TXD6 (transmit data output) is to be used, set bit 1 (P15SF1) to "1" to configure that port as a secondary function output. If the baud rate clock is to be input externally, specify with bit 2 (P15SF2) whether the input will be pulled-up.

(3) SIO6 transmit control register (ST6CON)
Reset bit 0 (ST6MOD) to "0" to change the mode to UART mode. Specify the transmit data length with bit 1 (ST6LN). Specify the stop bit length with bit 3 (ST6STB). Specify whether there is parity with bit 4 (ST6PEN). If parity is selected, specify the parity bit logic with bit 5 (ST6ODD). With bit 6 (TR6MIE), specify whether interrupt requests are enabled or disabled when a transmit buffer empty signal occurs. With bit 7 (TR6NIE), specify whether interrupt requests are enabled or disabled when a transmit complete signal occurs.

(4) SIO6 receive control register (SR6CON)
Specify with bit 2 (S6EXC) whether the baud rate clock is internal (overflow output of timer 3) or external (RXC6).

(5) SIO6 transmit-receive buffer register (S6BUF)
Transmission is started by writing the transmit data to S6BUF.

- **Receive settings**

(1) Port 15 mode register (P15IO)
If RXD6 (receive data input) is to be used, reset bit 0 (P15IO0) to "0" to configure that port as an input. If the baud rate clock is to be input externally, reset bit 2 (P15IO2) to "0" to configure that port as an input.

(2) Port 15 secondary function control register (P15SF)
Specify with bit 1 (P15SF0) whether the RXD6 pin will be pulled-up. If the baud rate clock is to be input externally, specify with bit 2 (P15SF2) whether the input will be pulled-up.

(3) SIO6 receive control register (SR6CON)

Reset bit 0 (SR6MOD) to "0" to change the mode to UART mode. Specify the receive data length with bit 1 (SR6LN). Specify with bit 2 (S6EXC) whether the baud rate clock is internal (overflow output of timer 3) or external (RXC6). Specify whether there is parity with bit 4 (SR6PEN). If parity is selected, specify the parity bit logic with bit 5 (SR6ODD). With bit 6 (RC6IE), specify whether interrupt requests are enabled or disabled when a receive complete signal occurs. If bit 7 (SR6REN) is set to "1", reception is enabled and the reception operation is performed when data arrives.

12.5.3.2 Synchronous Mode Settings

- **Transmit settings**

(1) Port 15 mode register (P15IO)

If TXD6 (transmit data output) is to be used, set bit 1 (P15IO1) to "1" to configure that port as an output. If the transmit clock is to be output externally (master mode), set bit 3 (P15IO3) to "1" to configure that port as an output. If the baud rate clock is to be input externally (slave mode), reset bit 3 (P15IO3) to "0" to configure that port as an input.

(2) Port 15 secondary function control register (P15SF)

If TXD6 (transmit data output) is to be used, set bit 1 (P15SF1) to "1" to configure that port as a secondary function output. If the transmit clock is to be output externally (master mode), set bit 3 (P15SF3) to "1" to configure that port as a secondary function output. If the baud rate clock is to be input externally (slave mode), specify with bit 3 (P15SF3) whether the input will be pulled-up.

(3) SIO6 transmit control register (ST6CON)

Set bit 0 (ST6MOD) to "1" to specify the mode to synchronous mode. Specify the transmit data length with bit 1 (ST6LN). Specify master or slave mode transmission with bit 3 (ST6STB). With bit 6 (TR6MIE), specify whether interrupt requests are enabled or disabled when a transmit buffer empty signal occurs. With bit 7 (TR6NIE), specify whether interrupt requests are enabled or disabled when a transmit complete signal occurs.

(4) SIO6 transmit-receive buffer register (S6BUF)

Transmission is started by writing the transmit data to S6BUF.

- **Receive settings**

(1) Port 15 mode register (P15IO)

If RXD6 (receive data input) is to be used, reset bit 0 (P15IO0) to "0" to configure that port as an input. If the transmit clock is to be output externally (master mode), set bit 2 (P15IO2) to "1" to configure that port as an output. If the transmit clock is to be input externally (slave mode), reset bit 2 (P15IO2) to "0" to configure that port as an input.

(2) Port 15 secondary function control register (P15SF)

Specify with bit 0 (P15SF0) whether the RXD6 pin will be pulled-up. If the transmit clock is to be output externally (master mode), set bit 2 (P15SF2) to "1" to configure that port as a secondary function output. If the transmit clock is to be input externally (slave mode), specify with bit 2 (P15SF2) whether the input will be pulled-up.

(3) SIO6 receive control register (SR6CON)

Set bit 0 (SR6MOD) to "1" to specify the mode to synchronous mode. Specify the receive data length with bit 1 (SR6LN). Specify the master or slave mode with bit 3 (SR6SLV). With bit 6 (RC6IE), specify whether interrupt requests are enabled or disabled when a receive complete signal occurs. If bit 7 (SR6REN) is set to "1", reception is enabled and the reception operation is performed when data arrives.

12.5.3.3 Baud Rate Generator (Timer 3) Settings

If overflow of timer 3 is selected for use as the baud rate clock, implement the following settings.

(1) General-purpose 8-bit timer 3 counter (TM3C)

Set the timer value that will be valid at the start of counting. When writing to TM3C, the same value will also be simultaneously and automatically written to the general-purpose 8-bit timer 3 register (TM3R).

(2) General-purpose 8-bit timer 3 control register (TM3CON)

Bits 0 to 2 (TM3C0 to TM3C2) of this register specify the count clock for timer 3. If bit 3 (TM3RUN) is set to "1", timer 3 will begin counting. If reset to "0", timer 3 will halt counting.

[Equation to Calculate Baud Rate]

$B = f_{(TM3)} \times 1/(256 - D) \times 1/n$

B : baud rate [bps]
$f_{(TM3)}$: timer 3 input clock frequency [Hz]
D : reload value (0 to 255)
n : 16 for the UART mode
4 for the synchronous mode

12.5.4 SIO6 Interrupt

When any SIO6 interrupt factor occurs, the interrupt request flag (QSIO6) is set to "1". The interrupt request flag (QSIO6) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ESIO6). The interrupt enable flag (ESIO6) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0SIO6 and P1SIO6). The interrupt priority setting flags (P0SIO6 and P1SIO6) are located in interrupt priority control register 6 (IP6).

Table 12-7 lists the vector address of the SIO6 interrupt factors and the interrupt processing flags.

Table 12-7 SIO6 Vector Address and Interrupt Processing Flags

<table>
<tr><td rowspan="2">Interrupt factor</td><td>Vector address [H]</td><td>Interrupt request</td><td>Interrupt enable</td><td colspan="2">Priority level</td></tr>
<tr><td></td><td></td><td></td><td>1</td><td>0</td></tr>
<tr><td>SIO6 transmit buffer empty signal is generated</td><td rowspan="3">003E</td><td rowspan="3">QSIO6</td><td rowspan="3">ESIO6</td><td rowspan="3">P1SIO6</td><td rowspan="3">P0SIO6</td></tr>
<tr><td>SIO6 transmit complete signal is generated</td></tr>
<tr><td>SIO6 receive complete signal is generated</td></tr>
<tr><td colspan="2" rowspan="2">Symbols (byte) of registers that contain interrupt processing flags</td><td>IRQ3</td><td>IE3</td><td colspan="2">IP6</td></tr>
<tr><td>Reference page</td><td>17-15</td><td>17-20</td><td colspan="2">17-28</td></tr>
</table>

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

12.6 SIO1, SIO6 Operation

12.6.1 Transmit Operation

- **UART mode**

Figure 12-10 shows the timing diagram of operation during UART transmission.

The clock pulse from the baud rate generator (timer 3 or timer 4) or from an external input is divided by 16 to generate the transmit shift clock.

If an external clock is to be used with the UART mode, input the clock to the receive clock I/O pin (RXCn) for SIOn. The externally input clock is processed as shown in figure 12-11, and is input to the 1/n dividing counter as the baud rate clock.

In synchronization with the transmit shift clock that has been generated, the transmission circuit controls transmission of the transmit data.

The SnBUF write signal (a signal that is output when an instruction to write to SnBUF is executed, for example "STB A, SnBUF") acts as a trigger to start transmission.

One CPU clock after the write signal is generated, transmit data in SnBUF is set in the transmit shift register. At this time, synchronized to the signal indicating the beginning of an instruction (M1S1), a transmit buffer empty signal is generated.

After the transmit data is set (after the fall of the data transfer signal to the transmit shift register), synchronized to the falling edge of the next transmit shift clock, the start bit is output from the transmit data output pin (TXDn). Thereafter, as specified by STnCON, the transmit data (LSB first), parity bit, and finally the stop bit are output to complete the transmission of one frame.

At this time, if the next transmit data has not been written to SnBUF, a transmit complete signal is generated in synchronization with M1S1, and the transmission is completed.

Because generation of the transmit shift clock is always unrelated to writes to SnBUF, from the time when transmit data is written to SnBUF until the start bit is output, there is a delay of a maximum of 16 baud rate clocks.

Because each of SIO1 and SIO6 has SnBUF and the transmit shift register which are designed in a duplex construction, during a transmission it is possible to write the next transmit data to SnBUF. If SnBUF is written to during a transmission, after the current one frame transmission is completed, the next transmit data will be automatically set in the transmit shift register, and the data transmission will continue. After one frame of data is transmit, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

Figure 12-14 shows the timing diagram of operation during continuous transmission.

- **Synchronous mode (SIO1, SIO6)**

[Master mode]

Figure 12-12 shows the timing diagram of operation during master mode transmission.

The clock pulse from the baud rate generator (timer 4 for SIO1 timer 3 for SIO6) is divided by 4 to generate the transmit shift clock.

In synchronization with the transmit shift clock that has been generated, the transmission circuit controls transmission of the transmit data.

The SnBUF write signal (the signal that is output when an instruction to write to SnBUF is executed, for example "STB A, SnBUF") acts as a trigger to start transmission.

One CPU clock after the write signal is generated, transmit data in SnBUF is set in the transmit shift register. At this time, synchronized to the signal indicating the beginning of an instruction (M1S1), a transmit buffer empty signal is generated.

After the transmit data is set (after the fall of the data transfer signal to the transmit shift register), synchronized to the falling edge of the next transmit shift clock, the external output clock begins to be output from the transmit clock I/O pin (TXCn). At the same time, transmit data is output LSB first from the transmit data output pin (TXDn). Thereafter, as specified by STnCON and synchronized to the transmit shift clock, transmit data is output to complete the transmission of one frame.

At this time, if the next transmit data has not been written to SnBUF, a transmit complete signal is generated in synchronization with M1S1, and the transmission is completed.

TXDn changes at the falling edge of TXCn. Therefore, at the receive side, TXDn is fetched at the rising edge of TXCn.

Because generation of the transmit shift clock is always unrelated to writes to SnBUF, from the time when transmit data is written to SnBUF until the first data is output, there is a delay of a maximum of 4 baud rate clocks.

Because each of SIO1 and SIO6 has SnBUF and the transmit shift register which are designed in a duplex construction, during a transmission it is possible to write the next transmit data to SnBUF. If SnBUF is written to during a transmission, after the current one frame transmission is completed, the next transmit data will be automatically set in the transmit shift register, and the data transmission will continue. After one frame of data is transmit, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

Figure 12-14 shows the timing diagram of operation during continuous transmission.

[Note]

During continuous transmission, there is a time lag of 1 bit between the current data transmission and the next data transmission, in which to set the next data. During this interval, TXDn is forced to a High level.

[Slave mode]

Figure 12-13 shows the timing diagram of operation during slave mode transmission.

In the slave mode, the transmit clock is input from the transmit clock I/O pin (TXCn). This external input clock is detected with the edge of CPU clock to generate the transmit shift clock.

In synchronization with the transmit shift clock that has been generated, the transmission circuit controls transmission of the transmit data.

The SnBUF write signal (the signal that is output when an instruction to write to SnBUF is executed, for example "STB A, SnBUF") acts as a trigger to start transmission.

One CPU clock after the write signal is generated, transmit data in SnBUF is set in the transmit shift register. At this time, synchronized to the signal indicating the beginning of an instruction (M1S1), a transmit buffer empty signal is generated.

After the transmit data is set (after the fall of the data transfer signal to the transmit shift register), synchronized to the falling edge of the next transmit shift clock, the transmit data is output LSB first from the transmit data output pin (TXDn). Thereafter, as specified by STnCON and synchronized to the transmit shift clock, transmit data is output to complete the transmission of one frame.

At this time, if the next transmit data has not been written to SnBUF, a transmit complete signal is generated in synchronization with M1S1, and the transmission is completed.

TXDn changes at the falling edge of the transmit shift clock that has been generated from the detected edge of the externally input TXCn. Therefore, at the receive side, TXDn is fetched at the rising edge of TXCn.

Because each of SIO1 and SIO6 has SnBUF and the transmit shift register which are designed in a duplex construction, during a transmission it is possible to write the next transmit data to SnBUF. If SnBUF is written to during a transmission, after the current one frame transmission is completed, the next transmit data will be automatically set in the transmit shift register, and the data transmission will continue. After one frame of data is transmit, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

Figure 12-14 shows the timing diagram of operation during continuous transmission.

[Note]

During continuous transmission, there is a time lag of 2 CPU clocks between the current data transmission and the data next transmission, in which to set the next data. During this interval, TXDn is forced to a High level. If an external clock is supplied, insert a margin of 2 or more CPU clocks between the current data transmission and the next data transmission.

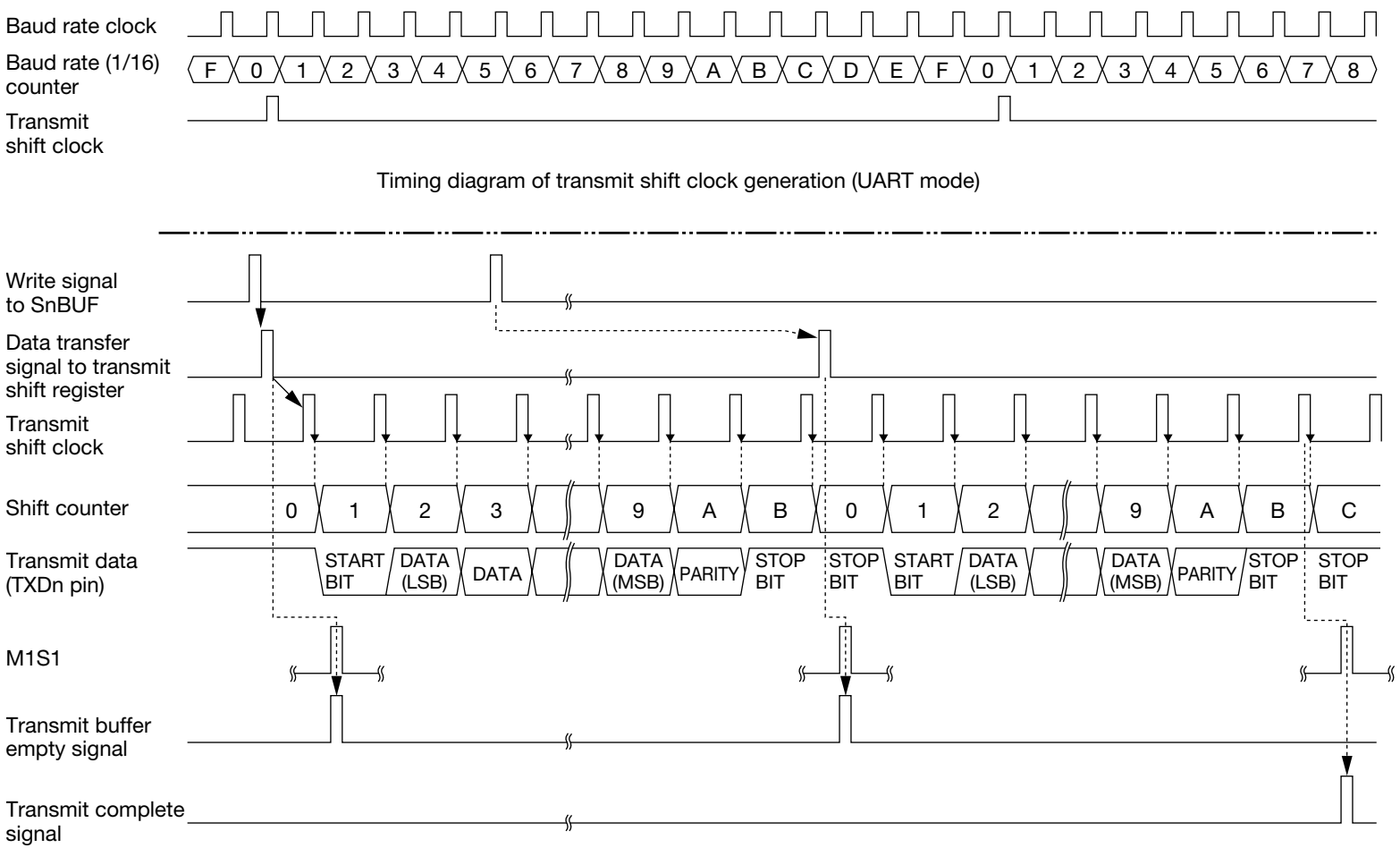


Figure 12-10 Transmission Timing Diagram (UART Mode)

Figure 12-11 Timing Diagram of Baud Rate Clock Generation by External Clock (UART Mode, Transmission and Reception)

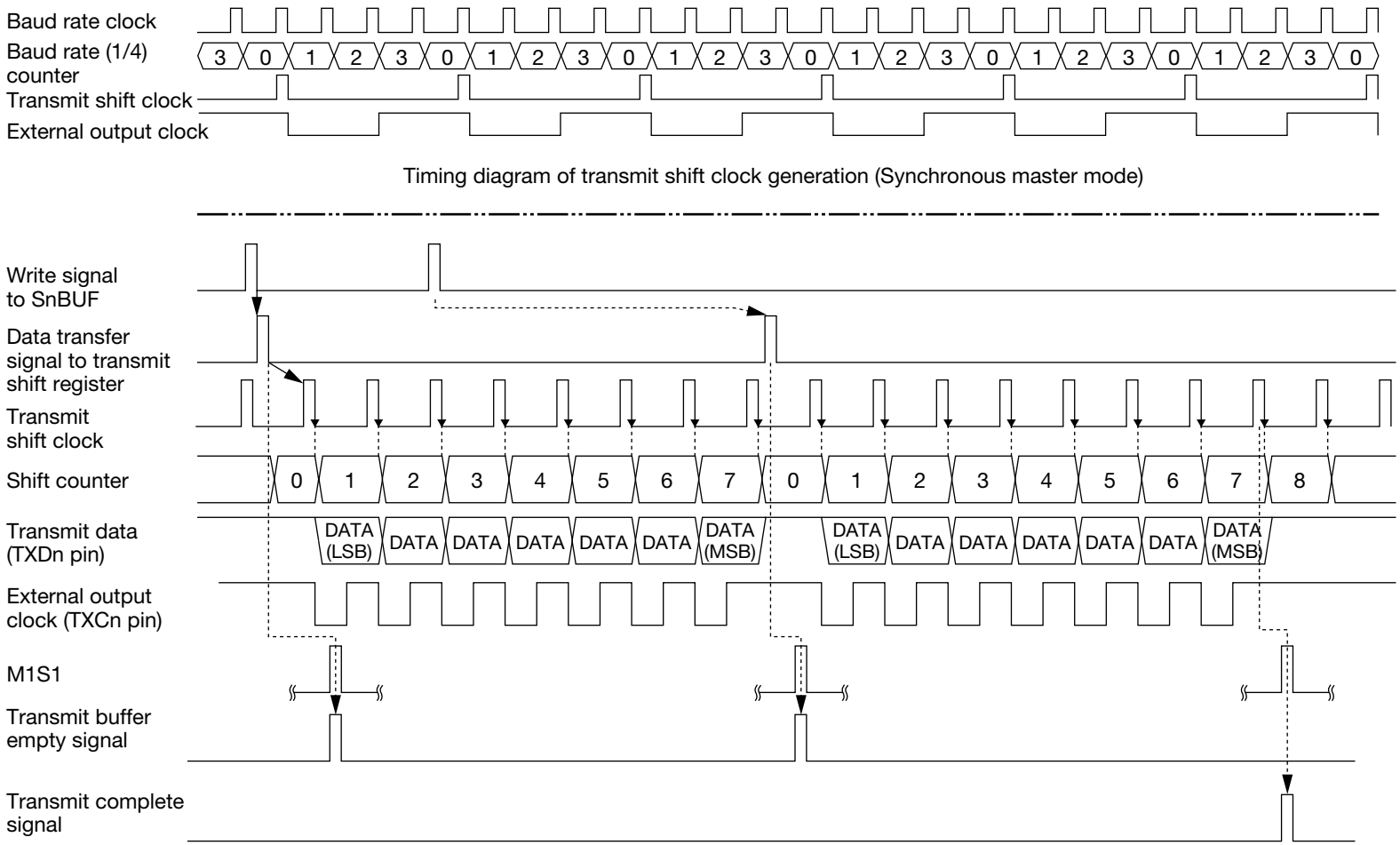
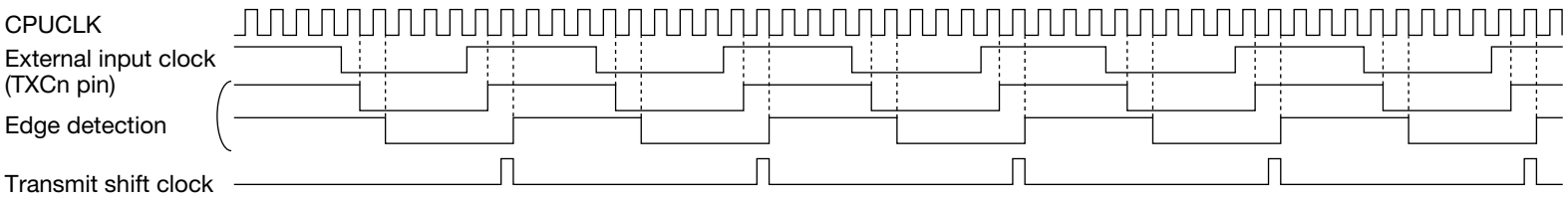


Figure 12-12 Transmission Timing Diagram (Synchronous Master Mode)



Timing diagram of transmit shift clock generation (Synchronous slave mode)

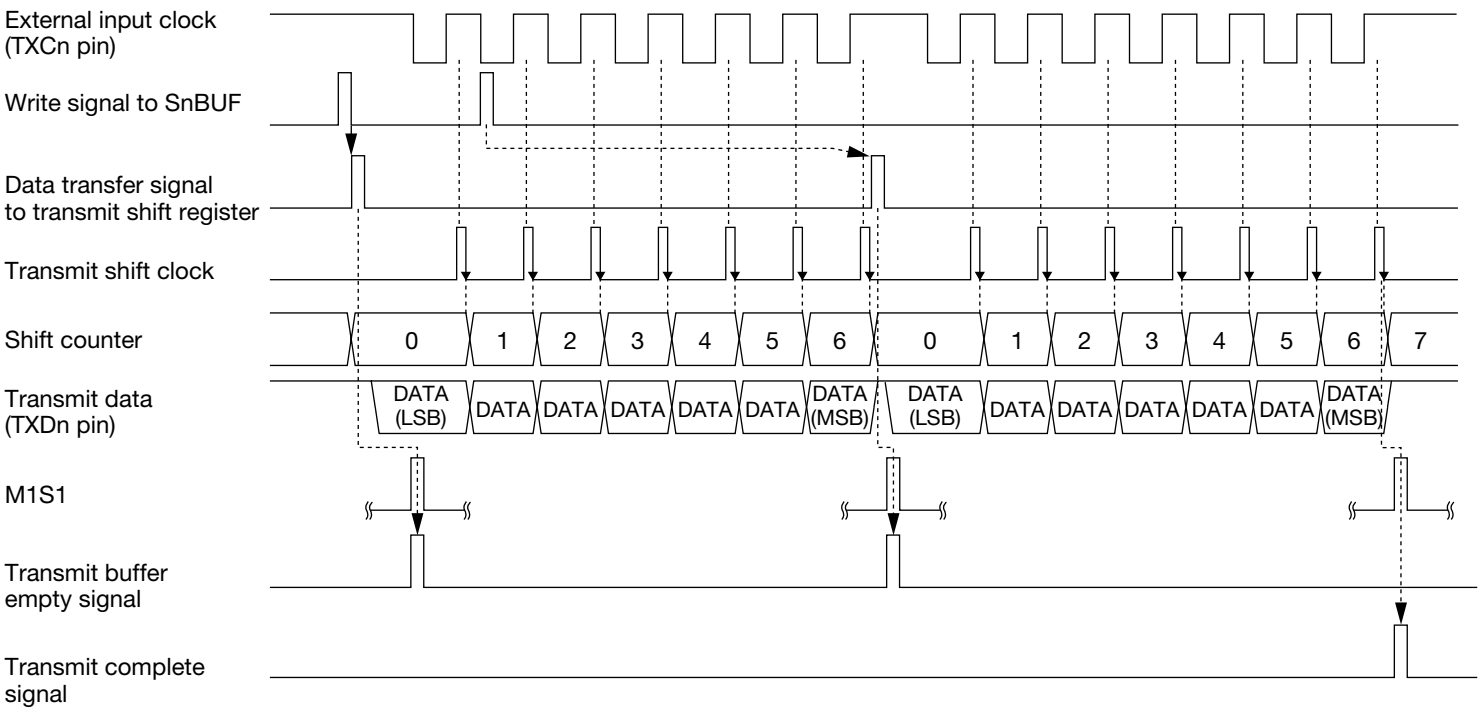


Figure 12-13 Transmission Timing Diagram (Synchronous Slave Mode)

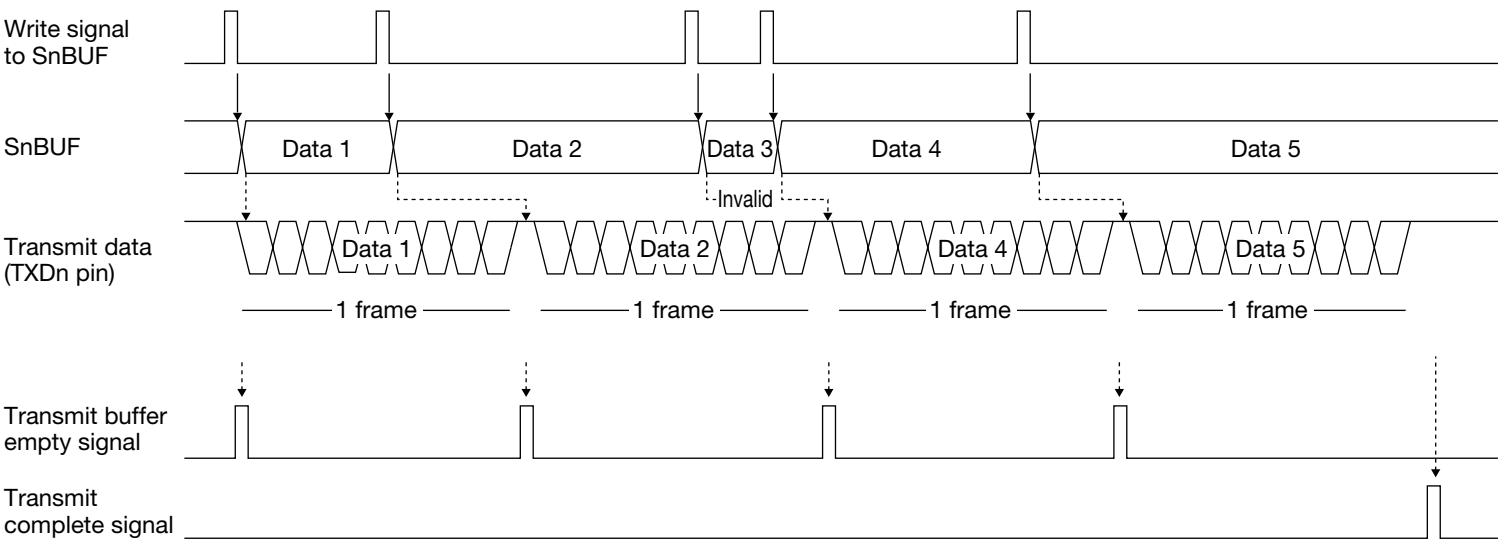


Figure 12-14 Transmission Timing Diagram (During Continuous Transmission)

12.6.2 Receive Operation

- **UART mode**

Figure 12-15 shows the timing diagram of operation during UART reception.

The clock pulse from the baud rate generator (timer 3 or timer 4) or from an external input is divided by 16 to generate the shift clock.

If an external clock is to be used with the UART mode, input the clock to the receive clock I/O pin (RXCn) for SIOn. The externally input clock is processed as shown in figure 12-11, and is input to the 1/n dividing counter as the baud rate clock.

The 1/n dividing circuit remains halted in its reset state until reception begins. The 7th, 8th and 9th pulses of the 1/16 divider (values 6, 7 and 8 of the baud rate (1/16) counter in figure 12-15) become the sampling clock for the receive data input pin (RXDn). The 10th pulse (value 9 of the baud rate (1/n) counter in figure 12-15) becomes the receive shift clock.

In synchronization with the receive shift clock, the reception circuit controls reception of the receive data.

A change in the receive data input pin (RXDn) from a High to Low level triggers the reception operation to start (at this time, SRnREN (bit 7) of SRnCON should be "1").

If the input signal to the receive data input pin (RXDn) is detected to have changed from a High to Low level, the 1/16 dividing counter that had been halted in its reset state now begins to operate. The start bit (L level) is sampled at the three sampling clocks of the 7th, 8th, and 9th pulses from the 1/16 dividing counter. If the start bit is at a Low level for two or more samples, it is judged to be valid. If not, the start bit is judged invalid, reception operation is initialized and then halted.

In a similar manner, receive data is sampled at the 7th, 8th, and 9th pulses from the 1/16 dividing counter. Data that is judged valid is shifted by the 10th clock, or in other words, by the receive shift clock, into the receive shift register as receive data. Thereafter, data reception continues as specified by SRnCON. The first stop bit (the 1st bit in the case of 2 stop bits) is received and the reception of one frame is completed.

At this time, if the received stop bit is "0", a framing error is issued. If the parity is incorrect, a parity error is issued. And, if the previously received data has not been read, an overrun error is issued (the previously received data will be overwritten).

However, at this time, the status register (SnSTAT) is not be updated of the detected error. Later, the contents of the receive shift register are transferred to SnBUF, a receive complete signal is generated in synchronization with M1S1 that indicates the beginning of the next instruction, and at the same timing, the status register (SnSTAT) is updated by the receive complete signal and each error signal. The series of receptions is completed.

- **Synchronous mode**

[Master mode]
Figure 12-16 shows the timing diagram of operation during master mode reception.

The clock pulse from the baud rate generator (timer 4 for SIO1 and timer 3 for SIO6) is divided by 4 to generate the external output clock. The 3rd pulse of the 1/4 divider (value 2 of the baud rate (1/4) counter in figure 12-16) becomes the sampling clock for the receive data input pin (RXDn). The 4th pulse (value 3 of the baud rate (1/4) counter in figure 12-16) becomes the receive shift clock.

In synchronization with the receive shift clock, the reception circuit controls reception of the receive data.

The falling edge of the receive shift clock immediately after SRnREN (bit 7) of SRnCON is set to "1" triggers the reception operation to start and the external output clock is output from the receive clock I/O pin (RXCn). At the next receive shift clock, receive data that was sampled at the prior sampling clock is shifted into the receive shift register.

At the falling edge of the external output clock, the transmit side transmits data. That data is shifted into the receive side at the falling edge of the transmit shift clock. Receive data is sampled only once. Thereafter, data reception continues as specified by SRnCON. After the last receive shift clock is output, the contents of the receive shift register are transferred to SnBUF, and a receive complete signal is generated in synchronization with M1S1, the signal that indicates the beginning of an instruction. At this time, an overrun error will be generated if the previously received data has not been read (the previously received data will be overwritten).

Finally, SRnREN of SRnCON is automatically cleared to "0" to complete the reception series.

[Slave mode]
Figure 12-17 shows the timing diagram of operation during slave mode reception.

In the slave mode, the receive clock is input externally (from the receive clock I/O pin (RXCn)). This external input clock is detected with the edge of CPU clock to generate the receive shift clock.

In synchronization with the receive shift clock that has been generated, the reception circuit controls receiving the receive data.

Reception operation is triggered to begin when SRnREN (bit 7) of SRnCON is set to "1" and the external input clock is input to the receive clock I/O pin (RXCn).

While the external input clock is at a Low level, the value of the receive data input pin (RXDn) is sampled. The sampled receive data is shifted into the receive shift register at the next receive shift clock. Thereafter, data reception continues as specified by SRnCON. After the last receive data is shifted in, the contents of the receive shift register are transferred to SnBUF, and a receive complete signal is generated in synchronization with M1S1, the signal that indicates the beginning of an instruction. At this time, an overrun error will be generated if the previously received data has not been read (the previously received data will be overwritten). This completes a one frame reception.

In the slave mode, SRnREN is not automatically cleared to "0" after completing the reception. If the receive shift clock continues to be input, the receive operation will restart.

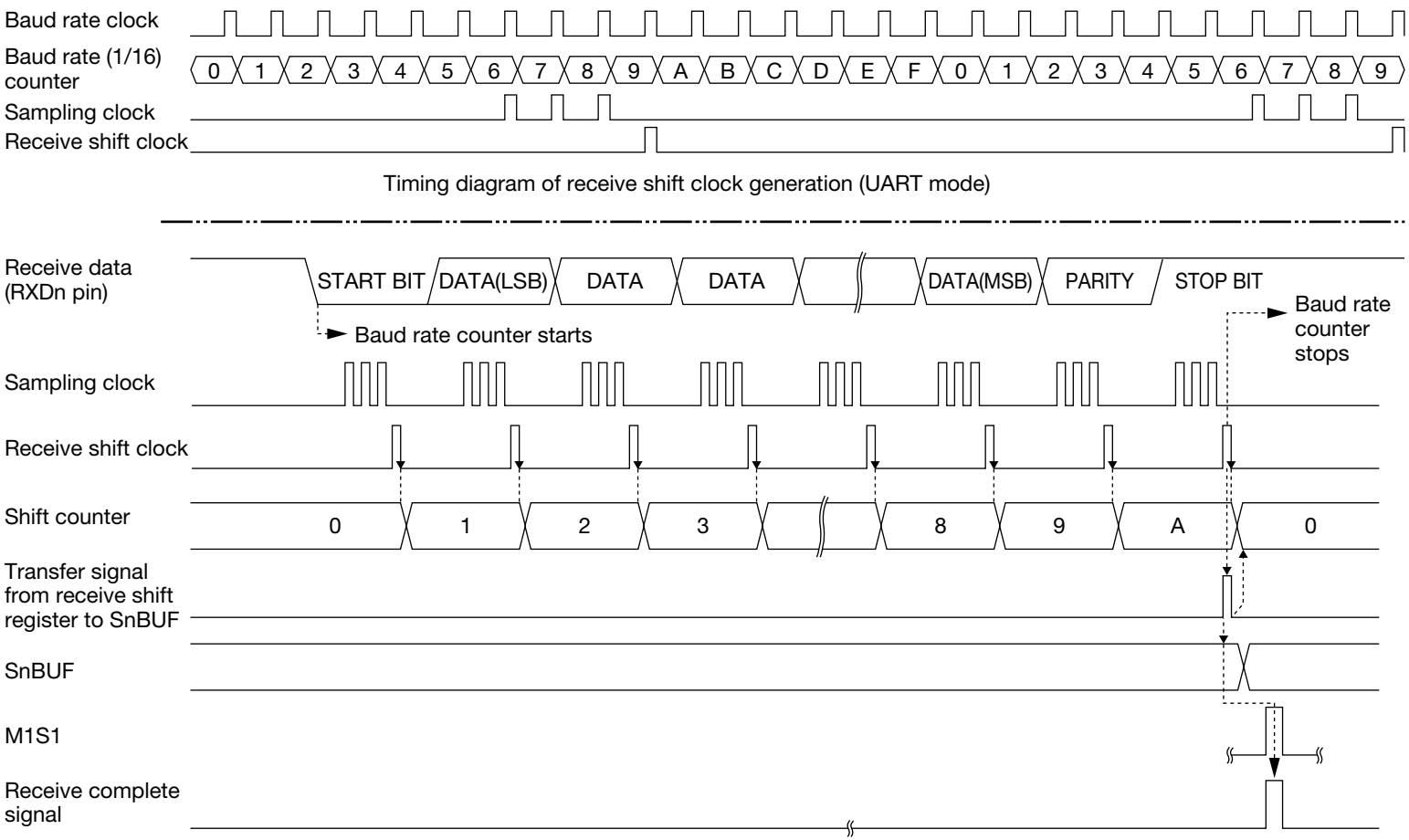


Figure 12-15 Reception Timing Diagram (UART Mode)

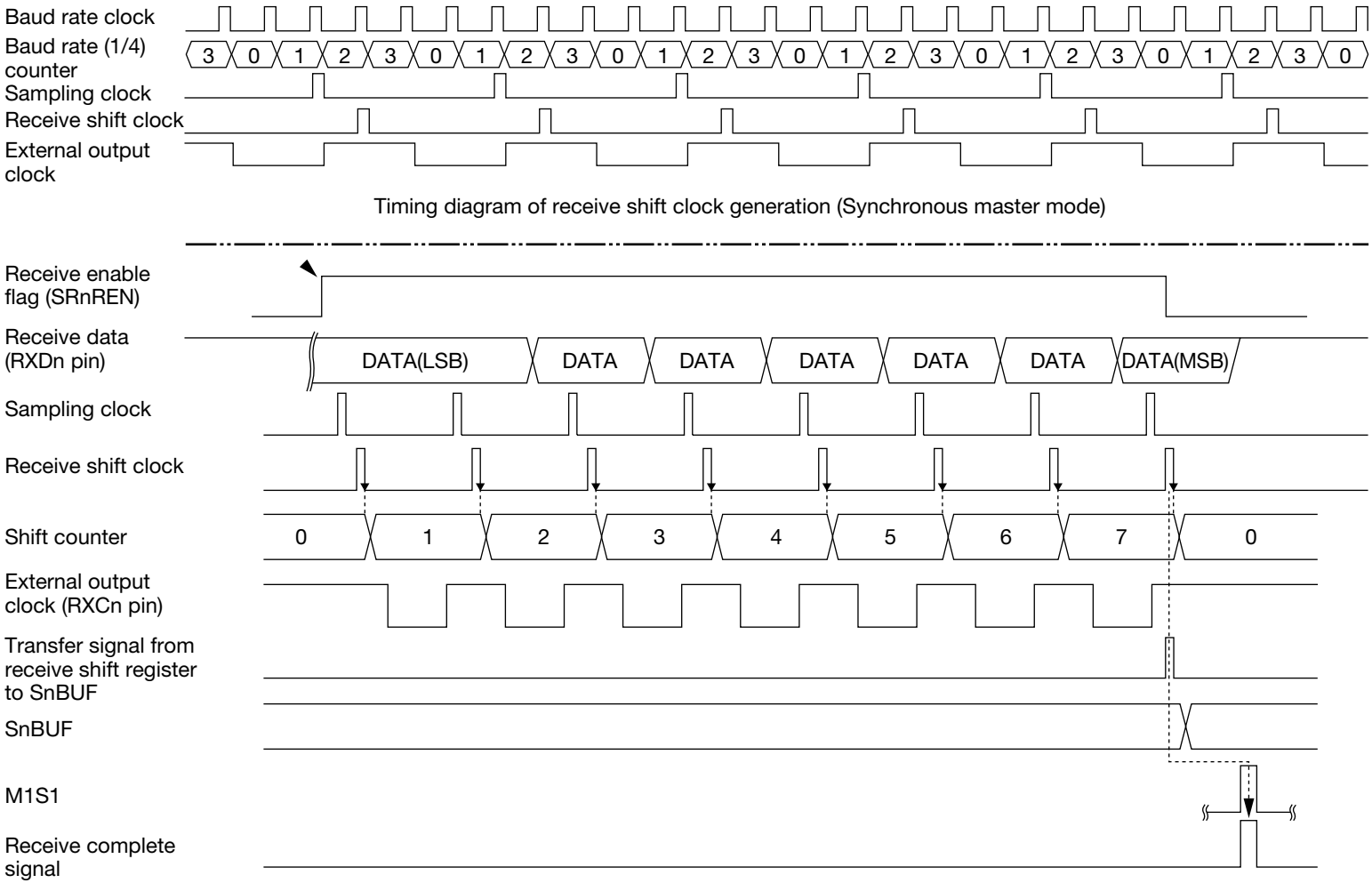


Figure 12-16 Reception Timing Diagram (Synchronous Master Mode)

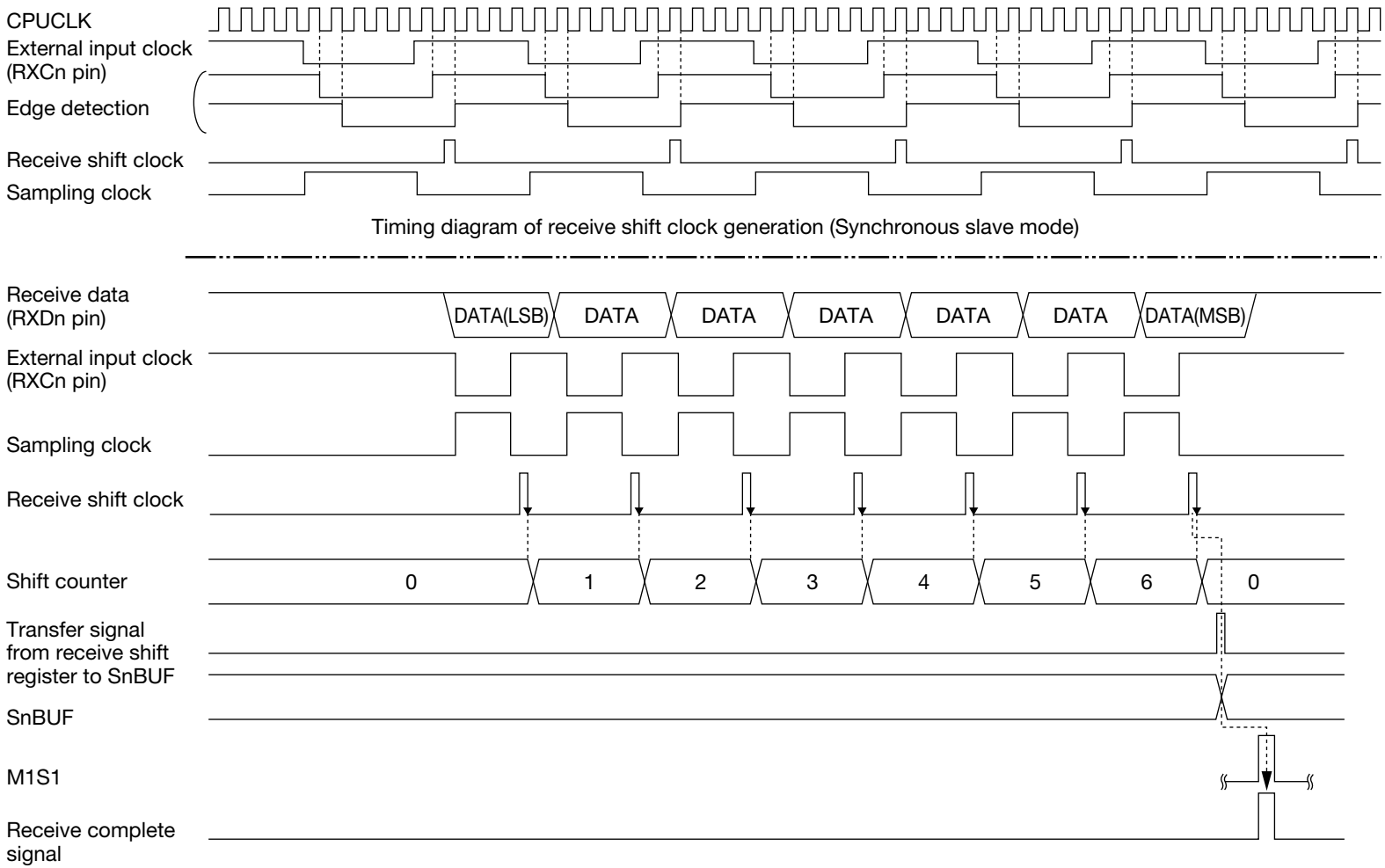


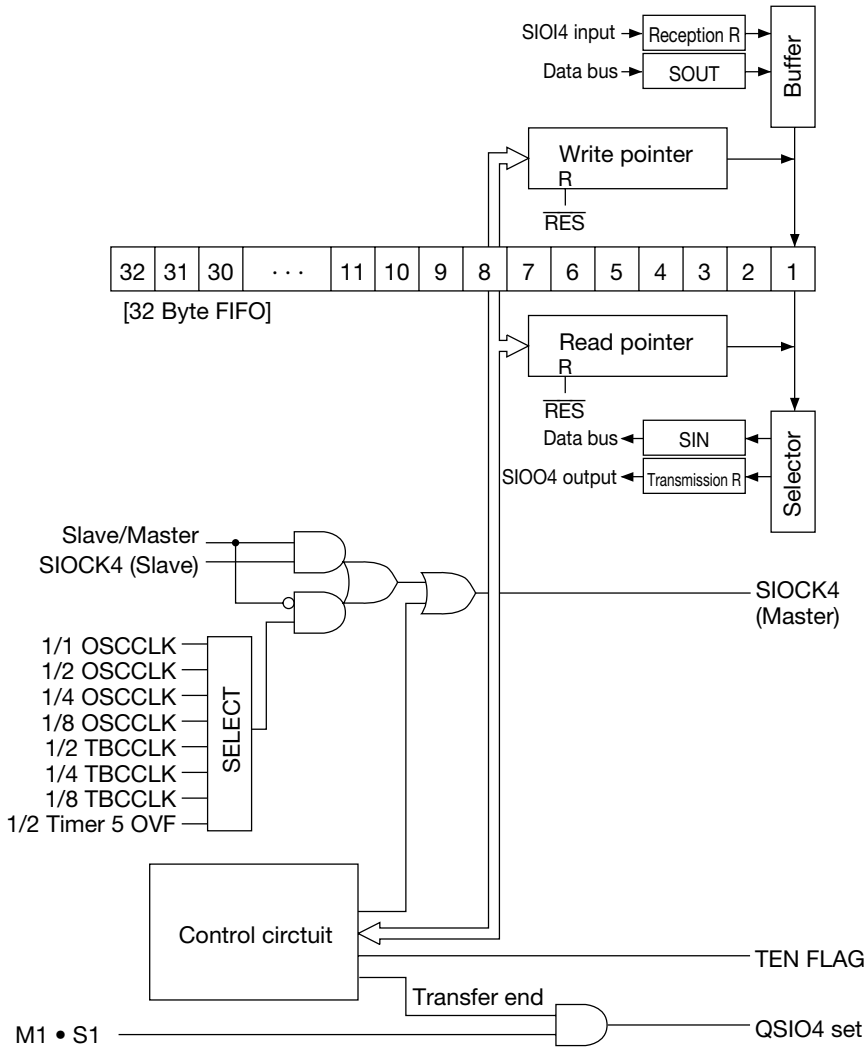
Figure 12-17 Reception Timing Diagram (Synchronous Slave Mode)

12.7 SIO4

SIO4 is an 8-bit auto transfer serial port used for clocked synchronous communication. Synchronized to the clock specified by the SIO4 control register (SIO4CON), SIO4 continuously transmits and receives 8 bits of data LSB first. When transmission and reception are complete, a transmit-receive interrupt is requested.

12.7.1 SIO4 Configuration

Figure 12-18 shows the SIO4 configuration.



SIO4CON: SIO4 control register
FIFOCON: FIFO control register
SIN4: SIO4 serial input FIFO data register
SOUT4: SIO4 serial output FIFO data register
SIOCK4: SIO4 transmit-receive clock input pin (P10_3)
SIOO4: SIO4 transmit data output pin (P10_4)
SIOI4: SIO4 receive data input pin (P10_5)

Figure 12-18 SIO4 Configuration

12.7.2 Description of SIO4 Registers

(1) SIO4 control register (SIO4CON)

The SIO4 control register (SIO4CON) is an 8-bit register that controls SIO4 operation.

SIO4CON can be read from and written to by the program. However, write operations are invalid for bit 7. If read, a value of "1" will always be obtained for bit 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SIO4CON becomes 80H.

Figure 12-19 shows the SIO4CON configuration.

[Description of each bit]

- SIO4C0 to SIO4C2 (bits 0 to 2)
  During the master mode, SIO4C0 to SIO4C2 select SIO4 clock. In the slave mode, these bits are invalid.
- SIO4SL (bit 3)
  SIO4SL specifies master or slave operation of SIO4.
- TEN4 (bit 4)
  When TEN4 is set to "1", transmission and reception begin. When transmission and reception are completed, it is automatically reset to "0".
- ICK4 (bit 5)
  ICK4 specifies whether there is a SIO4 interval clock. (Only valid during the master mode)
- BUSY4 (bit 6)
  BUSY4 indicates a transfer operation status. This can be used to determine the waiting time from setting TEN4 to "1" to transmission and reception start at multi-byte continuous transfer in the B mode of FIFO mode selection using SIO4 and SIO5 alternatively.

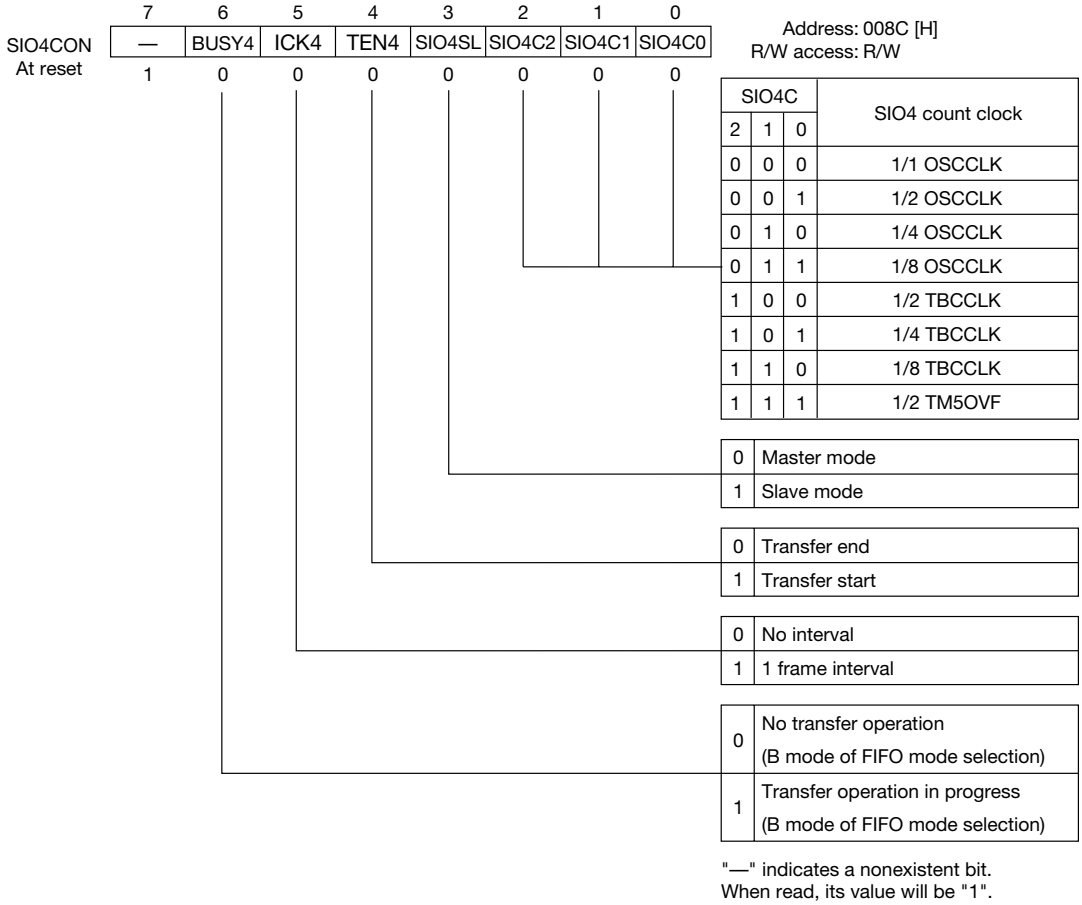


Figure 12-19 SIO4CON Configuration

(2) FIFO control register (FIFOCON)

The FIFO control register (FIFOCON) controls operation of the FIFO registers that are internal to SIO4 and SIO5.

FIFOCON can be read from and written to by the program. However, write operations to bits 0, 1, 4 and 5 are invalid.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of FIFOCON becomes 11H.

Figure 12-20 shows the FIFOCON configuration.

[Description of each bit]

- EMP4 (bit 0)
  EMP4 indicates the empty status of the SIO4's FIFO register. Due to SRES operation and at reset, the FIFO is cleared and enters the empty state. If all data in the FIFO register is read, the empty state is entered.
- FUL4 (bit 1)
  FUL4 indicates the full status of the SIO4's FIFO register. If 32 bytes of data are completely stored in the FIFO register, the FIFO full state (FUL = 1) is entered.
- ORE4 (bit 2)
  ORE4 indicates the overflow status of the SIO4's FIFO register (only valid during the slave mode). After completing reception of the number of bytes that were written to the FIFO register before the transfer, if an external clock is input, ORE4 is set to "1". In this case, because the FIFO register contents cannot be guaranteed, it is necessary to transfer the data again. ORE4 can be reset to "0" by setting SRE4 (bit 3) to "1".
- SRE4 (bit 3)
  SRE4 initializes SIO4. If SRE4 is set to "1", SIO4 will be initialized. After initialization, SRE4 is automatically reset to "0".
- EMP5 (bit 4)
  EMP5 indicates the empty status of the SIO5's FIFO register. Due to SRES operation and at reset, the FIFO is cleared and enters the empty state. If all data in the FIFO register is read, the empty state is entered.
- FUL5 (bit 5)
  FUL5 indicates the full status of the SIO5's FIFO register. If 32 bytes of data are completely stored in the FIFO register, the FIFO full state (FUL = 1) is entered.
- ORE5 (bit 6)
  ORE5 indicates the overflow status of the SIO5's FIFO register (only valid during the slave mode). After completing reception of the number of bytes that ware written to the FIFO register before the transfer, if an external clock is input, ORE5 is set to "1". In this case, because the FIFO register contents cannot be guaranteed, it is necessary to transfer the data again. ORE5 can be reset to "0" by setting SRE5 (bit 7) to "1".
- SRE5 (bit 7)
  SRE5 initializes SIO5. If SRE5 is set to "1", SIO5 will be initialized. After initialization, SRE5 is automatically reset to "0".

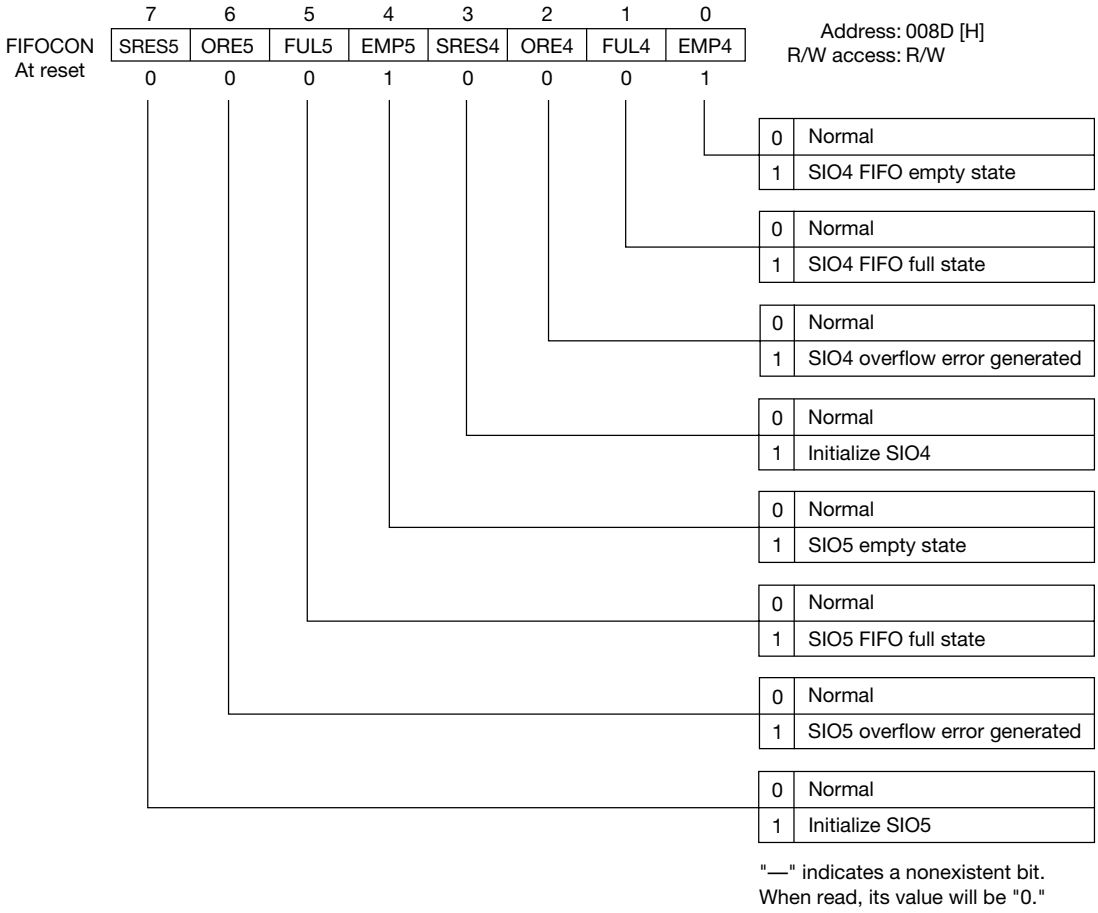


Figure 12-20 FIFOCON Configuration

(3) Serial input FIFO data register (SIN4)

The serial input FIFO data register (SIN4) is used to read 8-bit data received from the SIO pin. Since SIN4 is read-only, do not attempt to write to this register.

When 1 byte of received data has been gathered in the shift register, it is automatically loaded into the FIFO register. When transfer of the specified number of bytes is complete, an interrupt is generated. After the interrupt is generated, by reading SIN4, data can be read in order from the earliest received data. Because incorrect transmission or reception will occur if SIN4 is read during serial transmission or reception, do not attempt to read SIN4 while a transmission or reception is in progress.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of SIN4 are undefined.

(4) Serial output FIFO data register (SOUT4)

The serial output FIFO data register (SOUT4) is used to write the 8-bit serial data to be output from the SIOO4 pin. Since SOUT4 is write-only, do not attempt to read this register.

After data written to the SOUT4 register has been stored in the FIFO register, the start of transmission or reception causes that data to be sequentially loaded into a shift register.

Because incorrect transmission or reception will occur if SOUT4 is written to during serial transmission or reception, do not attempt to write to SOUT4 while a transmission or reception is in progress.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of SOUT 4 are undefined.

12.7.3 Example of SIO4-related Register Settings

- **Master mode settings**

(1) Port 10 mode register (P10IO)

If SIOCK4 (clock output) is to be used, set bit 3 (P10IO3) to "1" to configure the port as an output. Also if SIOO4 (transmit data output) is to be used during transmission, set bit 4 (P10IO4) to "1" to configure the port as an output, and if SIOI4 (receive data input) is to be used during reception, reset bit 5 (P10IO5) to "0" to configure the port as an input.

(2) Port 10 secondary function control register (P10SF)

If SIOCK4 (clock output) is to be used, set bit 3 (P10IO3) to "1" to configure the port as a secondary function output. Also, if SIOO4 (transmit data output) is to be used during transmission, set bit 4 (P10SF4) to "1" to configure the port as a secondary function output, and if SIOI4 (receive data input) is to be used during reception, reset bit 5 (P10SF5) to "0" to configure the port as a secondary function input.

(3) Serial output FIFO data register (SOUT4)

Write transmit data to SOUT4 (serial output FIFO data register).

[Note]

Writing to SOUT4 register and reading SIN4 are disabled during transmission or reception. It is necessary to write dummy data of transmission bytes beforehand for only reception, and it is necessary to read dummy data for transmission bytes after transmission only for transmission.

(4) SIO4 control register (SIO4CON)

Set SIO4 clock with bits 0 to 2 (SIO4C0 to SIO4C2). Reset bit 3 (SIO4SL) to "0" to set master mode. Specify whether there is an interval clock with bit 5 (ICK4). Transmission and reception are started by setting bit 4 (TEN4) to "1".

- **Slave mode settings**

(1) Port 10 mode register (P10IO)

If SIOCK4 (clock output) is to be used, reset bit 3 (P10IO3) to "0" to configure the port as an input. Also, if SIOO4 (transmit data output) is to be used during transmission, set bit 4 (P10IO4) to "1" to configure the port as an output, and if SIOI4 (receive data input) is to be used during reception, reset bit 5 (P10IO5) to "0" to configure the port as an input.

(2) Port 10 secondary function control register (P10SF)

If SIOCK4 (clock output) is to be used, reset bit 3 (P10IO3) to "0" to configure the port as a secondary function input. Also, if SIOO4 (transmit data output) is to be used during transmission, set bit 4 (P10SF4) to "1" to configure the port as a secondary function output, and if SIOI4 (receive data input) is to be used during reception, reset bit 5 (P10SF5) to "0" to configure the port as a secondary function input.

(3) Serial output FIFO data register (SOUT4)

Write transmit data to SOUT4 (serial output FIFO data register).

[Note]

Writing to SOUT4 register is disabled during transmission or reception. It is necessary to write dummy data of transmission bytes beforehand for only reception.

(4) SIO4 control register (SIO4CON)

SIO4 clock settings and specification of whether there is an interval clock with bit 5 (ICK4) are invalid. Set bit 3 (SIO4SL) to "1" to set slave mode. Transmission and reception are started by setting bit 4 (TEN4) to "1".

12.7.4 SIO4 Interrupt

When the SIO4 interrupt factor occurs, the interrupt request flag (QSIO4) is set to "1". The interrupt request flag (QSIO4) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ESIO4). The interrupt enable flag (ESIO4) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0SIO4 and P1SIO4). The interrupt priority setting flags (P0SIO4 and P1SIO4) are located in interrupt priority control register 6 (IP6).

Table 12-8 lists the vector address of the SIO4 interrupt factor and the interrupt processing flags.

Table 12-8 SIO4 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
SIO4 transmit-receive complete signal is generated	0040	QSIO4	ESIO4	P1SIO4	P0SIO4
Symbols (byte) of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	17-15	17-20	17-28	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

12.7.5 SIO4 Operation

SIO4 can select the master mode or slave mode, and can transfer a maximum of 32-byte transmit data continuyously.

In the master mode, the clock selected by bits 2 to 0 of SIO4CON is SIO4 clock. The SIO4 clock is output from the SIOCK4 pin.

In the slave mode, the clock input from the SIOCK4 pin is the SIO4 clock.

In both the master and slave modes, synchronized with the falling edge of the SIO4 clock, SIO4 outputs serial-out data from the SIOO4 pin. Synchronized with the rising edge of the SIO4 clock, serial-in data is input from the SIOI4 pin.

It is assumed that external devices change the serial-in data at the falling edge of the SIO4 clock and fetch the serial-out data at the rising edge of the SIO4 clock. Communication is executed in an LSB first mode.

Transfer operation is started by setting bit 4 (TEN4) of SIO4CON to "1" after writing transmit data to FIFO. When transfer is completed, bit TEN4 is reset to "0" and interrupt request flag (QSIO4) is set to "1" at the beginning of the next instruction (M1S1).

If bit TEN4 is reset to "0" during transfer, transmission and reception are immediately interrupted and SIO4 is initialized. The contents previously transferred are not assured. In the slave mode, set bit TEN4 to "1" when the SIOCK4 pin is at a high level to start transfer.

Figure 12-21 shows the timing of SIO4 operation during continuous transmission.

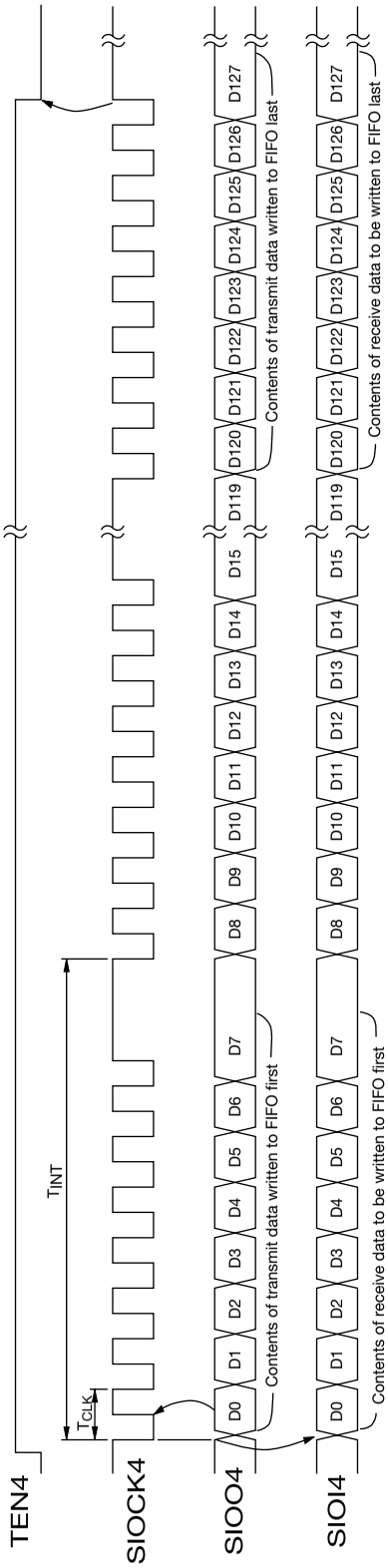


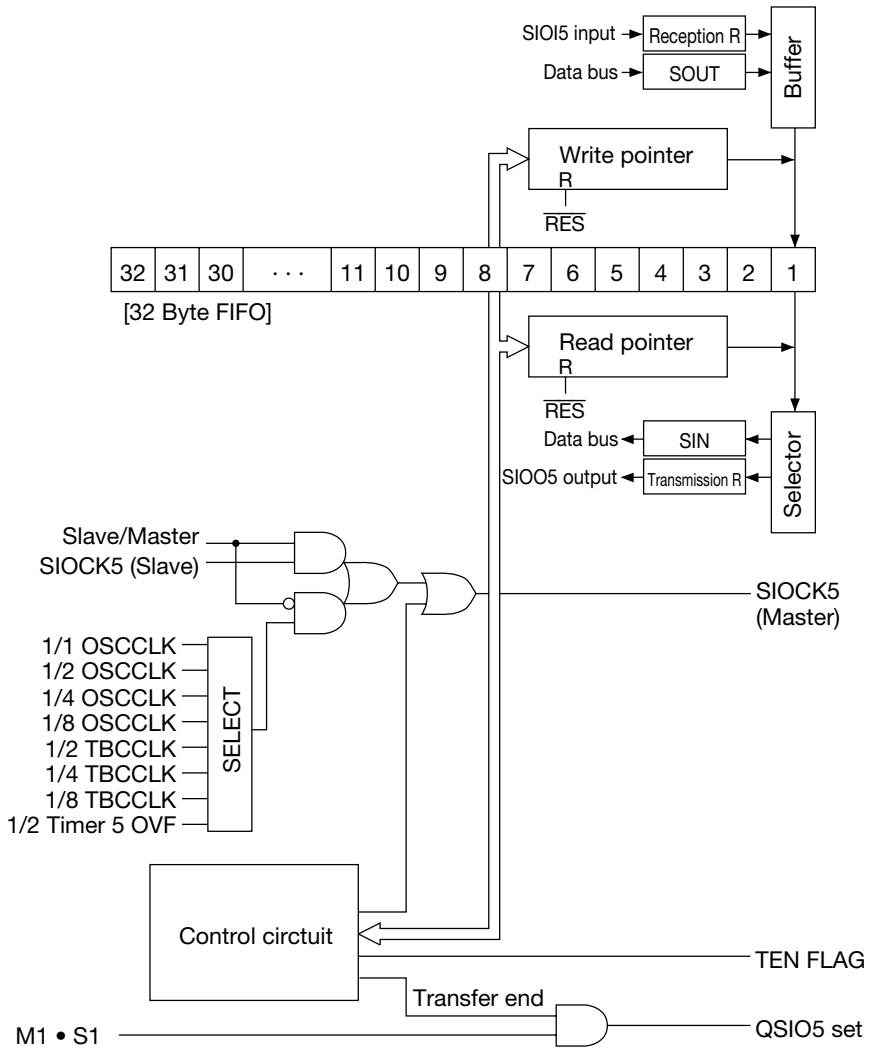
Figure 12-21 SIO4 Timing Diagram (During Continuous Transmission)

12.8 SIO5

SIO5 is an 8-bit auto transfer serial port used for clocked synchronous communication. Synchronized to the clock specified by the SIO5 control register (SIO5CON), SIO5 continuously transmits and receives 8 bits of data in an LSB first mode. When transmission and reception are complete, a transmit-receive interrupt is requested.

12.8.1 SIO5 Configuration

Figure 12-22 shows the SIO5 configuration.



SIO5CON: SIO5 control register
FIFOCON: FIFO control register
SIN5: SIO5 serial input FIFO data register
SOUT5: SIO5 serial output FIFO data register
SIOCK5: SIO5 transmit-receive clock input pin (P14_0)
SIOO5: SIO5 transmit data output pin (P14_1)
SIOI5: SIO5 receive data input pin (P14_2)

Figure 12-22 SIO5 Configuration

12.8.2 Description of SIO5 Registers

(1) SIO5 control register (SIO5CON)

The SIO5 control register (SIO5CON) is an 8-bit register that controls SIO5 operation.

SIO5CON can be read from and written to by the program. However, write operations are invalid for bit 7. If read, a value of "1" will always be obtained for bit 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SIO5CON becomes 80H.

Figure 12-23 shows the SIO5CON configuration.

[Description of each bit]

- SIO5C0 to SIO5C2 (bits0 to 2)
  During the master mode, SIO5C0 to SIO5C2 select SIO5 clock. In the slave mode, these bits are invalid.
- SIO5SL (bit 3)
  SIO5SL specifies master or slave operation of SIO5.
- TEN5 (bit 4)
  When TEN5 is set to "1", transmission and reception begin. When transmission and reception are completed, it is automatically reset to "0".
- ICK5 (bit 5)
  ICK5 specifies whether there is a SIO5 interval clock. (Only valid during the master mode)
- BUSY5 (bit 6)
  BUSY5 indicates a transfer operation status. This can be used to determine the waiting time from setting TEN5 to "1" to transmission and reception start at multi-byte continuous transfer using SIO4 and SIO5 alternatively.

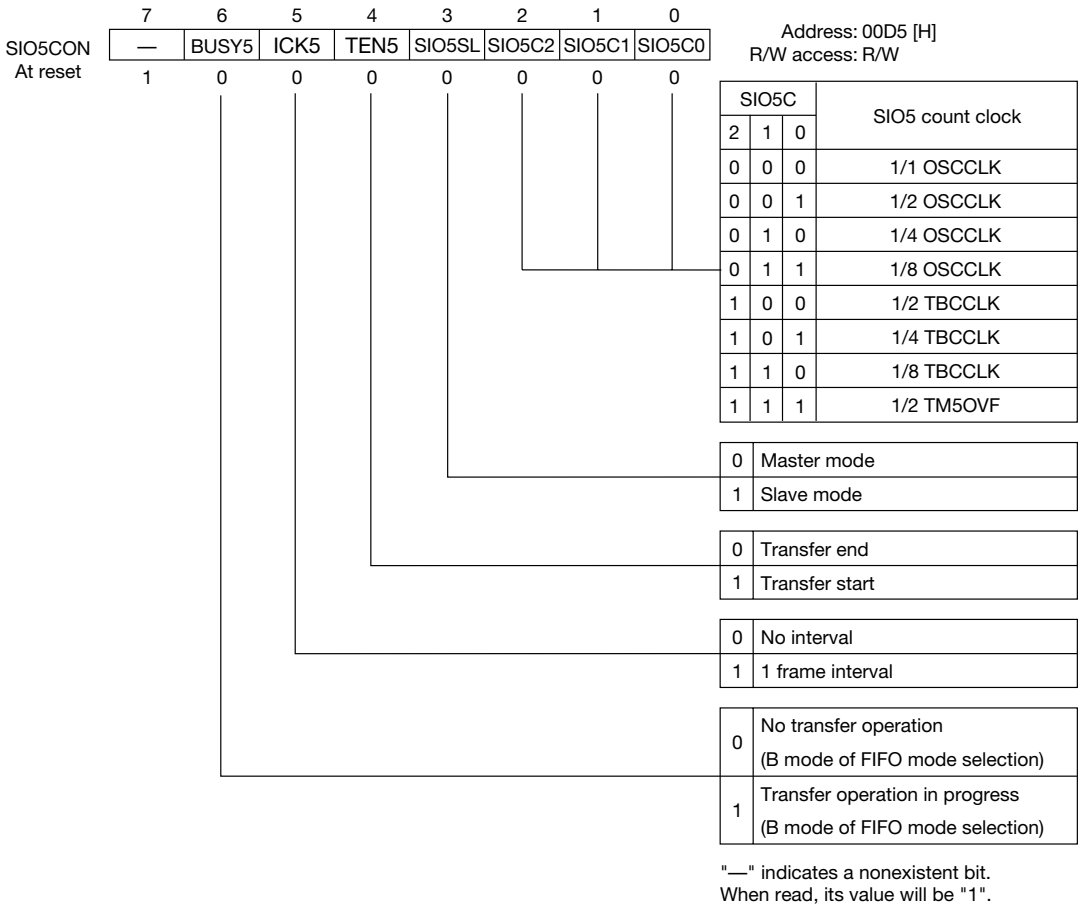


Figure 12-23 SIO5CON Configuration

(2) Serial input FIFO data register (SIN5)

The serial input FIFO data register (SIN5) is used to read 8-bit serial data received from the SIO pin. Since SIN5 is read-only, do not attempt to write to this register.

When 1 byte of received data has been gathered in the shift register, it is automatically loaded into the FIFO register. When transfer of the specified number of bytes is complete, an interrupt is generated. After the interrupt is generated, by reading SIN5, data can be read in order from the earliest received data. Because incorrect transmission or reception will occur if SIN5 is read during serial transmission or reception, do not attempt to read SIN5 while a transmission or reception is in progress.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of SIN5 are undefined.

(3) Serial output FIFO data register (SOUT5)

The serial output FIFO data register (SOUT5) is used to write the 8-bit serial data to be output from the SIOO5 pin. Since SOUT5 is write-only, do not attempt to read this register. After data written to the SOUT5 register has been stored in the FIFO register, the start of transmission or reception causes that data to be sequentially loaded into a shift register. Because incorrect transmission or reception will occur if SOUT5 is written to during serial transmission or reception, do not attempt to write to SOUT5 while a transmission or reception is in progress.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of SOUT5 are undefined.

(4) FIFO mode control register (FIFOMOD)

The FIFO mode control register (FIFOMOD) is a 2-bit register that specifies the mode of combined SIO4 and SIO5 usage. However, write operations to bits 2 through 7 are invalid. If bits 2 through 7 are read, a value of "1" will always be obtained.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), FIFOMOD becomes FCH.

Bits 0 and 1 (FMOD0, FMOD1) select the mode of combined SIO4 and SIO5 usage. Change these flags when neither SIO4 nor SIO5 is transferring data.

1) A mode

This mode operates SIO4 and SIO5 independently.

2) B mode

This mode alternates usage of SIO4 and SIO5 through the SIO4 port interface to consecutively transfer multiple bytes of data without limitation due to the number of FIFO stages. In this mode, operation of only master mode transmission and reception is possible. In this mode, while a transfer is in progress for one SIO, even if the TEN flag of the other SIO is set, that SIO's transfer will wait until the first SIO transfer is completed. After the first SIO transfer is completed, the other SIO transfer will automatically start. This provides for uninterrupted transfer.

In this mode, be sure to alternate usage of SIO4 and SIO5. For example, if SIO4→SIO5→SIO4 was used, then use SIO5→SIO4→SIO5 next. If B mode usage is intermixed with other modes, for example, if SIO5→SIO4→SIO5 (B mode) is used, after setting and using another mode, upon returning to the B mode, use SIO4→SIO5→SIO4 as the next sequence.

Interval clock settings (ICK) are invalid in this mode.

3) C mode

Via the SIO5 port interface, this mode can monitor the SIO4 transfer operation performed through the SIO4 interface.

SIO5 itself cannot be used with this mode.

4) D mode

Via the SIO4 port interface, this mode can monitor the SIO4 transfer operation performed through the SIO5 interface.

SIO5 itself cannot be used with this mode.

Using the C or D modes, two systems of port interfaces can be connected to SIO4. Using the A or D modes, two systems of SIOs (SIO4/SIO5) can be connected to the single SIO4 port interface.

Figure 12-24 shows the FIFOMOD configuration and Figure 12-25 shows the B, C, and D mode configurations.

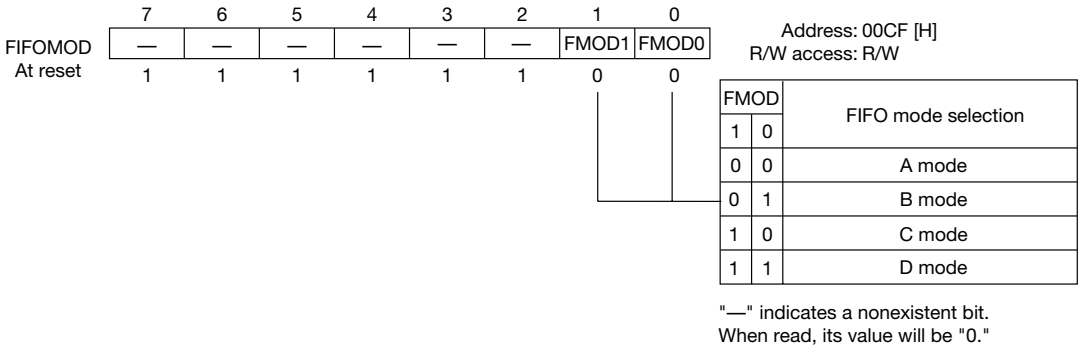
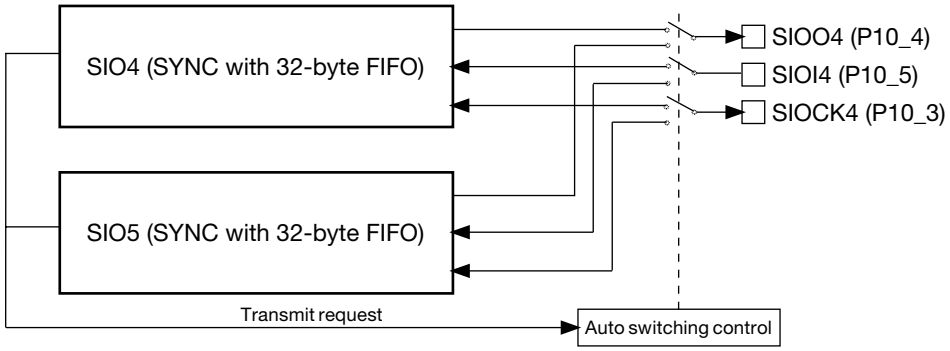
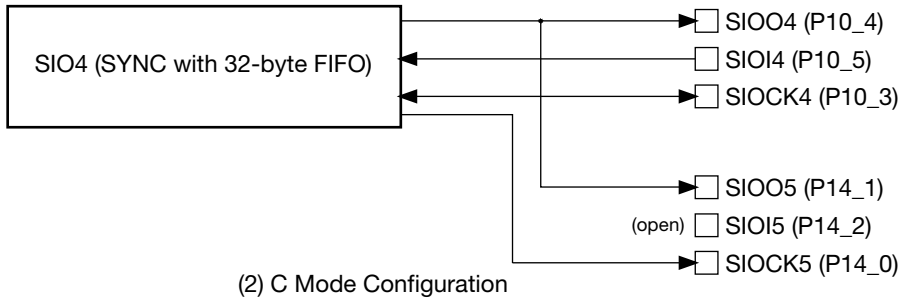


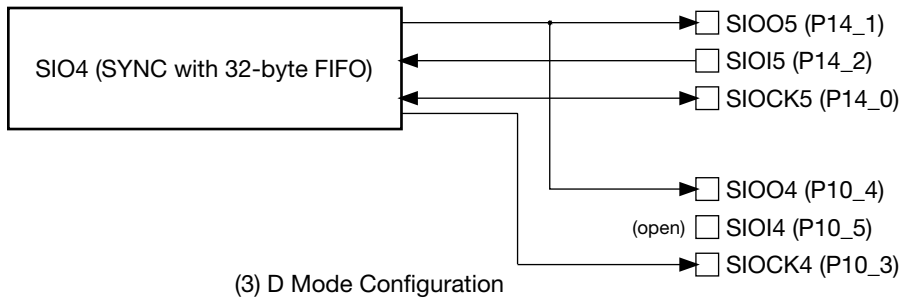
Figure 12-24 FIFOMOD Configuration



(1) B Mode Configuration



(2) C Mode Configuration



(3) D Mode Configuration

Figure 12-25 Mode Configuration

12.8.3 Example of SIO5-related Register Settings

- **Master mode settings**

(1) Port 14 mode register (P14IO)

If SIOCK5 (clock output) is to be used, set bit 0 (P14IO0) to "1" to configure the port as an output. Also, if SIOO5 (transmit data output) is to be used during transmission, set bit 1 (P14IO1) to "1" to configure the port as an output, and if SIOI5 (receive data input) is to be used during reception, reset bit 2 (P14IO2) to "0" to configure the port as an input.

(2) Port 14 secondary function control register (P14SF)

If SIOCK5 (clock output) is to be used, set bit 0 (P14IO0) to "1" to configure the port as a secondary function output. Also, if SIOO5 (transmit data output) is to be used during transmission, set bit 1 (P14SF1) to "1" to configure the port as a secondary function output, and if SIOI5 (receive data input) is to be used during reception, reset bit 2 (P14SF2) to "0" to configure the port as a secondary function input.

(3) Serial output FIFO data register (SOUT5)

Write transmit data to SOUT5 (serial output FIFO data register).

[Note]

Writing to SOUT5 register is disabled during transmission or reception. It is necessary to write dummy data of transmission bytes beforehand for only reception.

(4) SIO5 control register (SIO5CON)

Set SIO5 clock with bits 0 to 2 (SIO5C0 to SIO5C2). Reset bit 3 (SIO5SL) to "0" to set master mode. Specify whether there is an interval clock with bit 5 (ICK5). Transmission and reception are started by setting bit 5 (TEN5) to "1".

- **Slave mode settings**

(1) Port 14 mode register (P14IO)

If SIOCK5 (clock output) is to be used, reset bit 0 (P14IO0) to "0" to configure the port as an input. Also, if SIOO5 (transmit data output) is to be used during transmission, set bit 1 (P14IO1) to "1" to configure the port as an output, and if SIOI5 (receive data input) is to be used during reception, reset bit 2 (P14IO2) to "0" to configure the port as an input.

(2) Port 14 secondary function control register (P14SF)

If SIOCK5 (clock output) is to be used, reset bit 0 (P14IO0) to "0" to configure the port as a secondary function input. Also, if SIOO5 (transmit data output) is to be used during transmission, set bit 1 (P14SF1) to "1" to configure the port as a secondary function output, and if SIOI5 (receive data input) is to be used during reception, reset bit 2 (P14SF2) to "0" to configure the port as a secondary function input.

(3) Serial output FIFO data register (SOUT5)

Write transmit data to SOUT5 (serial output FIFO data register).

[Note]

Writing to SOUT5 register is disabled during transmission or reception. Reading SIN5 is also disabled. It is necessary to write dummy data of transmission bytes beforehand for only reception, and it is necessary to read dummy data for transmission bytes after transmission only for transmission.

(4) SIO5 control register (SIO5CON)

SIO5 clock settings and specification of whether there is an interval clock with bit 5 (ICK5) are invalid. Set bit 3 (SIO5SL) to "1" to set slave mode. Transmission and reception are started by setting bit 4 (TEN5) to "1".

12.8.4 SIO5 Interrupt

When the SIO5 interrupt factor occurs, the interrupt request flag (QSIO5) is set to "1". The interrupt request flag (QSIO5) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ESIO5). The interrupt enable flag (ESIO5) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0SIO5 and P1SIO5). The interrupt priority setting flags (P0SIO5 and P1SIO5) are located in interrupt priority control register 6 (IP6).

Table 12-9 lists the vector address of the SIO5 interrupt factor and the interrupt processing flags.

Table 12-9 SIO5 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
SIO5 transmit-receive complete signal is generated	003C	QSIO5	ESIO5	P1SIO5	P0SIO5
Symbols (byte) of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	17-15	17-20	17-28	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

12.8.5 SIO5 Operation

SIO5 can select the master mode or slave mode, and can transfer a maximum of 32-byte transmit data continuously.

In the master mode, the clock selected by bits 2 to 0 of SIO5CON is SIO5 clock. The SIO5 clock is output from the SIOCK5 pin.

In the slave mode, the clock input from the SIOCK5 pin is the SIO5 clock.

In both the master and slave modes, synchronized with the falling edge of the SIO5 clock, SIO5 outputs serial-out data from the SIOO5 pin. Synchronized with the rising edge of the SIO5 clock, serial-in data is input from the SIOI5 pin.

It is assumed that external devices change the serial-in data at the falling edge of the SIO5 clock and fetch the serial-out data at the rising edge of the SIO5 clock. Communication is executed in an LSB first mode.

Transfer operation is started by setting bit 4 (TEN5) of SIO5CON to "1" after writing transmit data to FIFO. When transfer is completed, bit TEN5 is reset to "0" and interrupt request flag (QSIO5) is set to "1" at the beginning of the next instruction (M1S1).

If bit TEN5 is reset to "0" during transfer, transmission and reception are immediately interrupted and SIO5 is initialized. The contents previously transferred are not assured. In the slave mode, set bit TEN5 to "1" when the SIOCK5 pin is at a high level to start transfer.

Figure 12-26 shows the timing of SIO5 operation during continuous transmission.

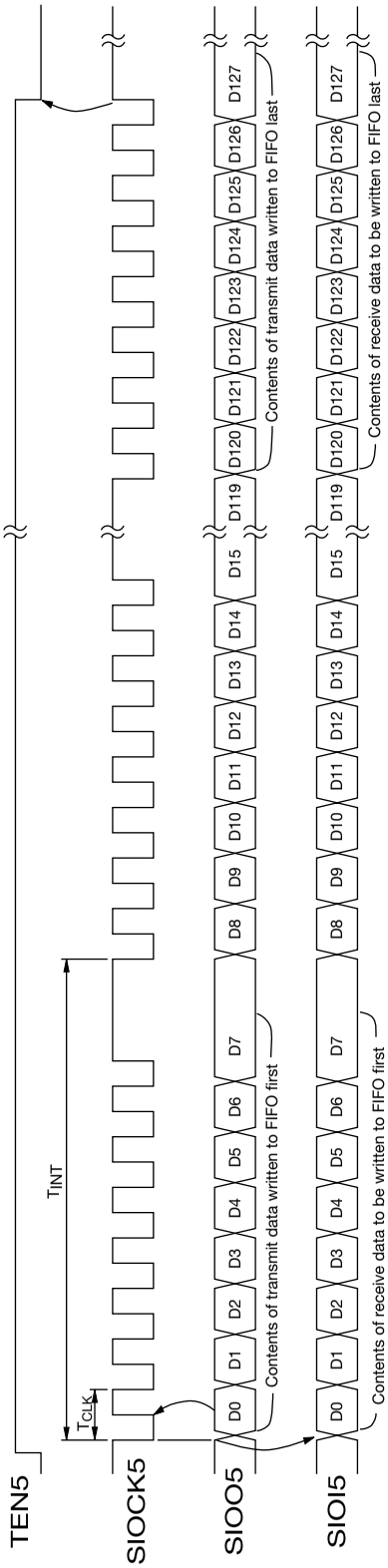


Figure 12-26 SIO5 Timing Diagram (During Continuous Transmission)

Chapter 13

A/D Converter Functions

13. A/D Converter Functions

13.1 Overview

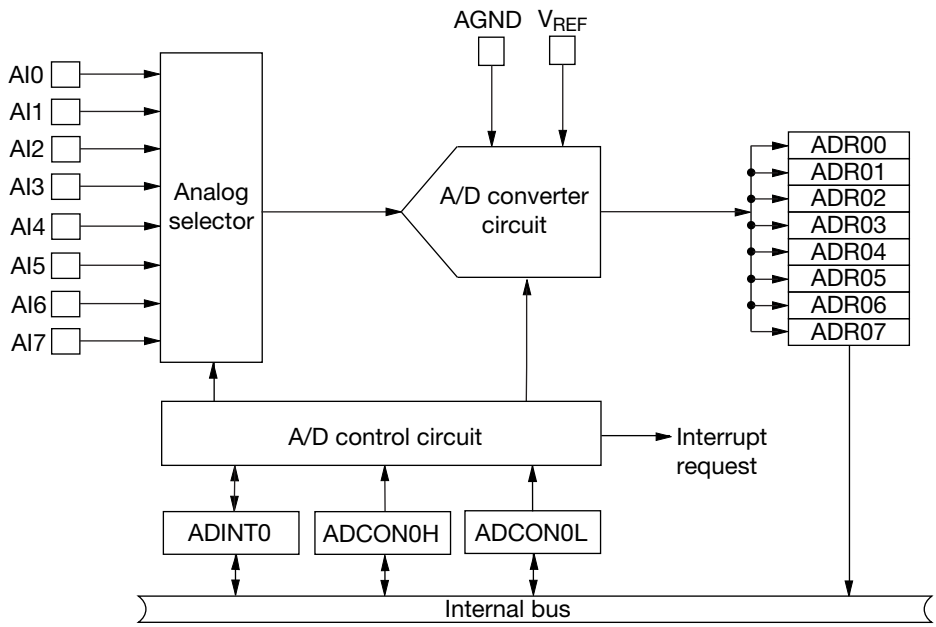
The MSM66577 family has an internal 8-channel A/D converter with 10-bit resolution.

The A/D converter can operate in a scan mode that sequentially converts several selected channels, or in a select mode that converts one selected channel.

A successive comparison method with a sample and hold function is used to convert analog quantities to digital quantities.

13.2 A/D Converter Configuration

Figure 13-1 shows the A/D converter configuration.



AI0 to AI7: analog input pins (P12_0 to P12_7)
ADR00 to ADR07: A/D result register (10 bits)
ADINT0: A/D interrupt control register 0
ADCON0H: A/D control register 0H
ADCON0L: A/D control register 0L
AGND: analog GND pin
V_{REF}: analog reference voltage pin

Figure 13-1 A/D Converter Configuration

13.3 A/D Converter Registers

Table 13-1 lists a summary of SFRs for control of the A/D converter.

Table 13-1 Summary of SFRs for A/D Converter Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
009C☆	A/D control register 0L	ADCON0L	—	R/W	8	80	13-3
009D☆	A/D control register 0H	ADCON0H	—	R/W	8	00	13-5
009E☆	A/D interrupt control register 0	ADINT0	—	R/W	8	F0	13-7
00A0	A/D result register 00	—	ADR00	R	16	Undefined	13-8
00A1		—					
00A2	A/D result register 01	—	ADR01	R	16	Undefined	13-8
00A3		—					
00A4	A/D result register 02	—	ADR02	R	16	Undefined	13-8
00A5		—					
00A6	A/D result register 03	—	ADR03	R	16	Undefined	13-8
00A7		—					
00A8	A/D result register 04	—	ADR04	R	16	Undefined	13-8
00A9		—					
00AA	A/D result register 05	—	ADR05	R	16	Undefined	13-8
00AB		—					
00AC	A/D result register 06	—	ADR06	R	16	Undefined	13-8
00AD		—					
00AE	A/D result register 07	—	ADR07	R	16	Undefined	13-8
00AF		—					

[Notes]

1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. Do not write to ADR00 through ADR07. If written to, the contents of all the registers from ADR00 through ADR07 may be overwritten.
4. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

13.3.1 Description of A/D Converter Registers

(1) A/D control register 0L (ADCON0L)

A/D control register 0L (ADCON0L) consists of 6 bits and specifies settings for the scan mode.

ADCON0L can be read from and written to by the program. However, write operations are invalid for bit 7. Also, if bit 3 is to be written to, a value of "0" must be written. If read, bit 3 is always "0" and bit 7 is always "1".

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ADCON0L becomes 80H.

Figure 13-2 shows the ADCON0L configuration.

[Description of each bit]

- ADSNM00 to ADSNM02 (bits 0 to 2)
  ADSNM00 to ADSNM02 specify the scan channels of the scan mode.
  Change the scan channels while the A/D converter is halted.
  Changes of the scan channels are valid only when ADRUN0 (bit 4) is "0".
- ADRUN0 (bit 4)
  ADRUN0 starts and stops A/D conversion in the scan mode.
  If set to "1", A/D conversion will begin. If reset to "0", conversion will be stopped. The ADRUN0 bit specifies to operate or to halt A/D conversion and is not a status flag indicating whether conversion is in progress or is halted.
- SNEX0 (bit 5)
  SNEX0 specifies the factor that activates A/D conversion in the scan mode.
  When SNEX0 is "0", after A/D conversion of the previous channel is complete, conversion of the next channel begins. When SNEX0 is "1", after A/D conversion of the previous channel is complete, 1 channel of A/D conversion is performed for each valid edge of the signal at the external interrupt input pin (EXINT1).
- SCNC0 (bit 6)
  SCNC0 specifies the operating mode after one cycle of scanning.
  When SCNC0 is "0", after one cycle of the specified scanning channels, A/D conversion starts again at the first channel.
  When SCNC0 is "1", after one cycle of the specified scanning channels, A/D conversion is stopped.
  If used in the "SCNC0 = 1" mode, A/D conversion is reactivated by resetting to "0" the INTSN0 flag that is located in ADINT0 and indicates when one cycle of scanning is complete. (Control with the ADRUN0 bit is unnecessary. With ADRUN0 set to "1", A/D conversion can be activated by resetting INTSN0 to "0".)
  If the mode is to be switched to "SCNC0 = 0" (the "after one cycle, start the next conversion" mode), reactivate the A/D conversion by resetting SCNC0 to "0". (Control with the ADRUN0 bit is unnecessary.)

[Note]

If used in the "after one cycle of scanning, stop the conversion" mode, A/D conversion can not be reactivated by resetting to "0" and then setting to "1" the ADRUN0 bit.

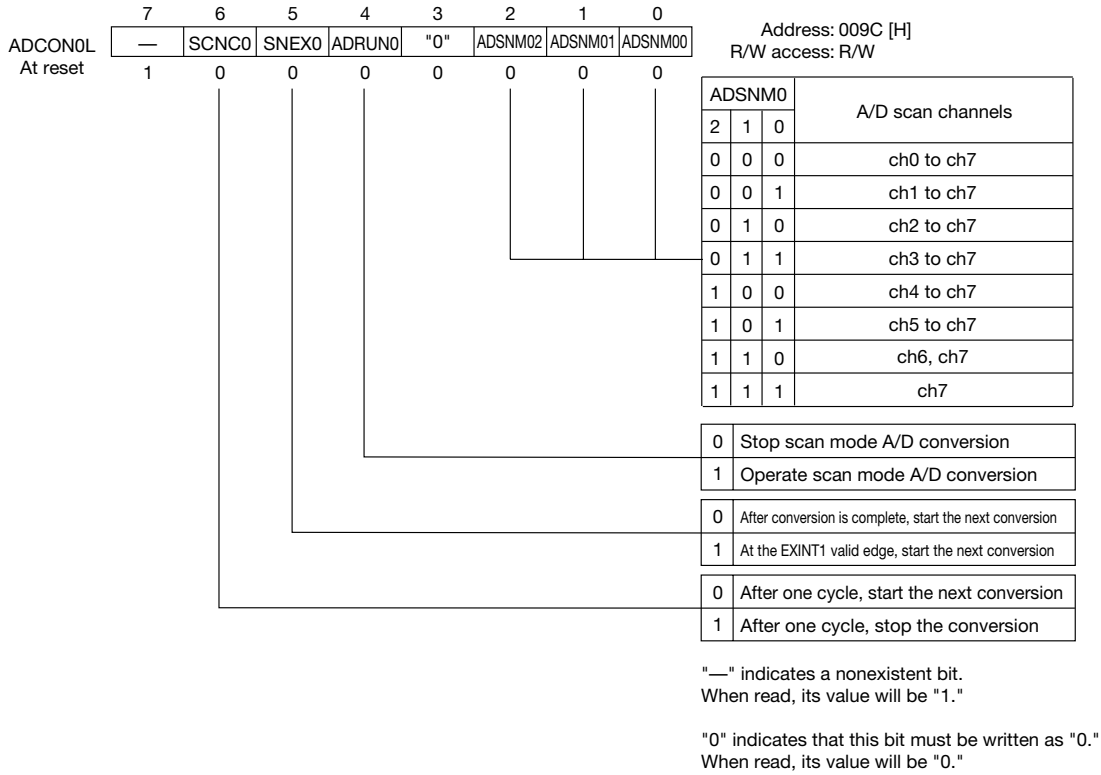


Figure 13-2 ADCON0L Configuration

(2) A/D control register 0H (ADCON0H)

ADCON0H is a 7-bit register that mainly controls the select mode of the A/D converter.

ADCON0H can be read from and written to by the program. However, if bit 3 is to be written to, a value of "0" must be written. If read, bit 3 is always "0".

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ADCON0H becomes 00H.

Figure 13-3 shows the ADCON0H configuration.

[Description of each bit]

- ADSTM00 to ADSTM02 (bits 0 to 2)
  ADSTM00 to ADSTM02 specify the A/D conversion channel of the select mode.
  Change the A/D conversion channel of the select mode while the A/D converter is halted.
  Changes of the conversion channel of the select mode are valid only when STS0 (bit 4) is "0".
- STS0 (bit 4)
  STS0 starts and stops A/D conversion in the select mode.
  If set to "1", A/D conversion will begin. If reset to "0", the conversion will be halted. When A/D conversion in the select mode is completed, STS0 is automatically reset to "0" by the hardware.
- ADTM00 to ADTM02 (bits 5 to 7)
  ADTM00 and ADTM01 specify the number of clocks required for the A/D conversion of 1 channel.
  Select an appropriate number of A/D conversion clocks based on the impedance of the analog input signal source and the frequency of the source.
  For further details, refer to Section 13.5, "Notes Regarding Usage of A/D Converter".
  During A/D conversion, changes to the number of clocks will be ignored.

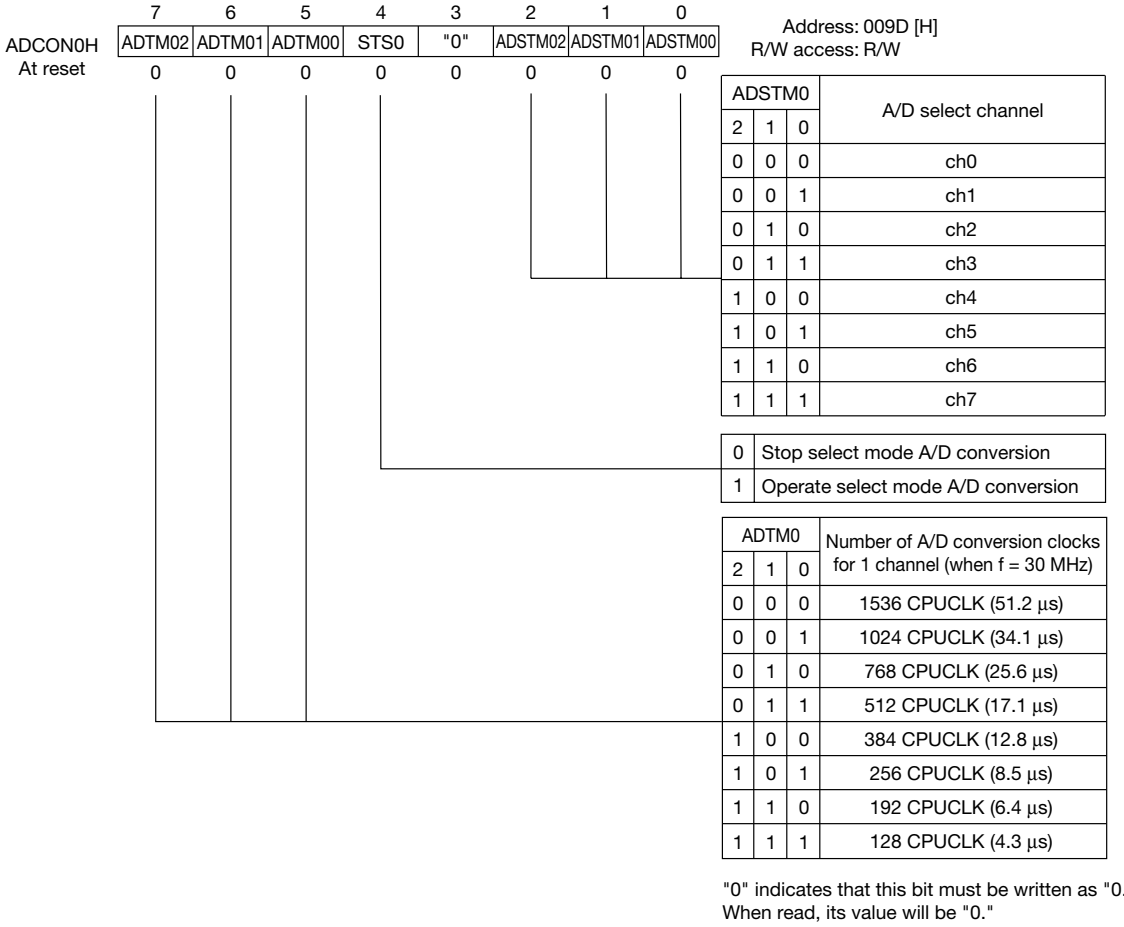


Figure 13-3 ADCON0H Configuration

(3) A/D interrupt control register (ADINT0)

ADINT0 is a 4-bit register that mainly controls the generation of interrupt requests by the A/D converter.

ADINT0 can be read from and written to by the program. However, write operations are invalid for bits 4 through 7. If read, a value of "1" will always be obtained for bits 4 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ADINT0 becomes F0H.

Figure 13-4 shows the ADINT0 configuration.

[Description of each bit]

- INTSN0 (bit 0)
  INTSN0 indicates whether one cycle of the scan channels has been completed.
  When INTSN0 is "0", then "one cycle is not complete". If "1", then "one cycle is complete".
  Here, "one cycle is complete" signifies that in the scan mode, A/D conversion of channel 7 is complete. INTSN0 must be reset to "0" by the program.
- INTST0 (bit 1)
  INTST0 indicates whether A/D conversion in the select mode is complete.
  When INTST0 is "1", then A/D conversion is complete. INTST0 must be reset to "0" by the program.
- ADSNIE0 (bit 2)
  ADSNIE0 enables or disables interrupt requests when one cycle of scan channels is complete. Here, "one cycle is complete" signifies that in the scan mode, A/D conversion of channel 7 is complete.
- ADSTIE0 (bit 3)
  ADSTIE0 enables or disables interrupt requests when A/D conversion is completed in the select mode.

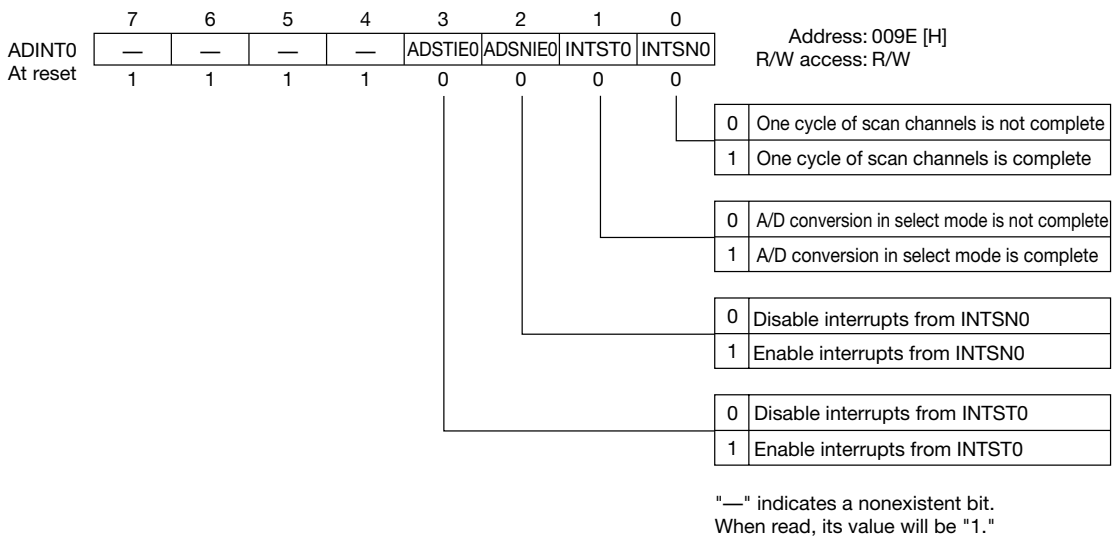


Figure 13-4 ADINT0 Configuration

(4) A/D result registers (ADR00 to ADR07)

A/D result registers (ADR00 to ADR07) consist of 10 bits and store the A/D conversion results.

A/D result registers (ADR00 to ADR07) can only be read in word access operations by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ADR00 to ADR07 is undefined.

Figure 13-5 shows the configuration of the A/D result registers (ADR00 to ADR07).

R/W access: R (word access only)

	Address [H]	7	6	5	4	3	2	1	0	
ADR07	00AF	—	—	—	—	—	—	bit9	bit8	bit9 : MSB
	00AE	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	bit0 : LSB
ADR06	00AD	—	—	—	—	—	—	bit9	bit8	
	00AC	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR05	00AB	—	—	—	—	—	—	bit9	bit8	
	00AA	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR04	00A9	—	—	—	—	—	—	bit9	bit8	
	00A8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR03	00A7	—	—	—	—	—	—	bit9	bit8	
	00A6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR02	00A5	—	—	—	—	—	—	bit9	bit8	
	00A4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR01	00A3	—	—	—	—	—	—	bit9	bit8	
	00A2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ADR00	00A1	—	—	—	—	—	—	bit9	bit8	
	00A0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

"—" indicates a nonexistent bit. When read, its value will be "0."

Figure 13-5 A/D Result Registers (ADR00 to ADR07) Configuration

[Note]

Do not write to the A/D result registers (ADR00 to ADR07). If written to, all the registers from ADR00 to ADR07 may be overwritten.

13.3.2 Example of A/D Converter-related Register Settings

- **Scan mode setting**

(1) A/D control register 0H (ADCON0H)
With bits 5 to 7 (ADTM00 to ADTM02), specify the number of clocks required for the A/D conversion per channel.

(2) A/D interrupt control register (ADINT0)
Specify that one cycle of the scan channels is not complete by resetting bit 0 (INTSN0) to "0". With bit 2 (ADSNIE0), enable or disable the generation of interrupts when one cycle of the scan channels is complete (INTSN0).

(3) A/D control register 0L (ADCON0L)
Specify the scan channels with bits 0 to 2 (ADSNM00 to ADSNM02). With bit 5 (SNEX0), specify the factor that will start A/D conversion. With bit 6 (SCNC0), specify operation after completion of one cycle of the scan channels. Set bit 4 (ADRUN0) to "1" to start the A/D conversion. If reset to "0", the A/D conversion can be stopped before completion.

- **Select mode setting**

(1) A/D interrupt control register (ADINT0)
Specify that the AD conversion in the select mode is not complete by resetting bit 1 (INTST0) to "0". With bit 3 (ADSTIE0), enable or disable the generation of interrupts when A/D conversion is completed in the select mode (INTST0).

(2) A/D control register 0H (ADCON0H)
Specify the A/D conversion channel with bits 0 to 2 (ADSTM00 to ADSTM02). With bits 5 to 7 (ADTM00 to ADTM02), specify the number of clocks required for the A/D conversion per channel. Set bit 4 (STS0) to "1" to start the A/D conversion. If reset to "0", the A/D conversion can be stopped before completion.

13.4 A/D Converter Operation

The A/D converter has two operating modes, the scan mode and the select mode.

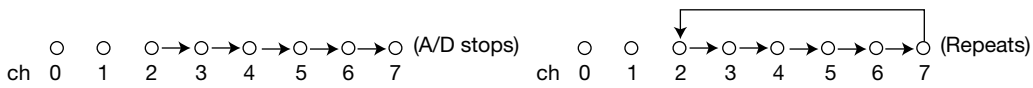
The scan mode sequentially performs A/D conversion of channels from an arbitrary channel to ch7. In the scan mode, when the A/D conversion of ch7 is complete, A/D conversion can be selected to either stop, or to automatically restart beginning at a specified channel.

Figure 13-6 shows an example of scan mode operation.

During scan mode operation, it is also possible to operate the select mode. In this case, when the select mode is activated, A/D conversion is halted for the channel being converted in scan mode, and A/D conversion of the specified channel is performed in the select mode. When the A/D conversion in the select mode is complete, scan mode A/D conversion is restarted for the channel that was previously halted.

The timing diagram of Figure 13-7 shows the select mode being executed during the scan mode.

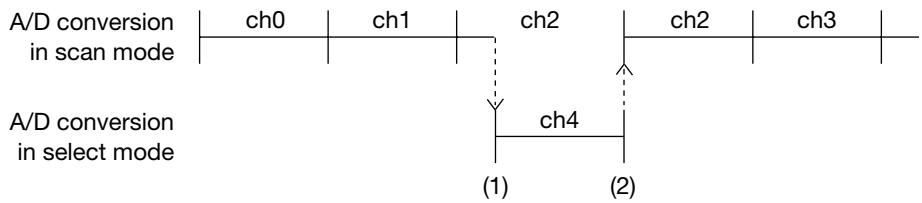
While the A/D converter is stopped and also during the STOP mode, the circuitry is controlled so that there is no current flow between V_{REF} and AGND. Therefore, it is not necessary to turn off the V_{REF} supply externally when it is not in use.



(a) Scan from ch2 to ch7
After one cycle, conversion stops (SCNC0 = 1)

(b) Scan from ch2 to ch7
After one cycle, next conversion starts (SCNC0 = 0)

Figure 13-6 Example Operation During Scan Mode



(1) Terminate A/D conversion of ch2
Start A/D conversion of ch4 in select mode

(2) Complete A/D conversion of ch4
Restart A/D conversion in scan mode beginning with ch2

Figure 13-7 Timing Diagram of Select Mode Execution During Scan Mode

13.5 Notes Regarding Usage of A/D Converter

13.5.1 Considerations When Setting the Conversion Time

Figure 13-8 shows an equivalent circuit of the analog input section of the A/D converter.

Because a successive comparison method with a sample and hold function is used in the converter, the internal sampling capacitor must be charged or discharged within a fixed sampling time to reach a voltage level that corresponds to the required precision.

The number of clocks required for the A/D conversion of 1 channel can be specified with ADTM00 and ADTM01 of the A/D control register 0H (ADCON0H).

Table 13-2 lists the clock allocation for the A/D conversion processes of 1 channel. Because the actual sampling time is determined by the operating frequency of the microcomputer, actual sampling times can be computed from the numeric values in this table.

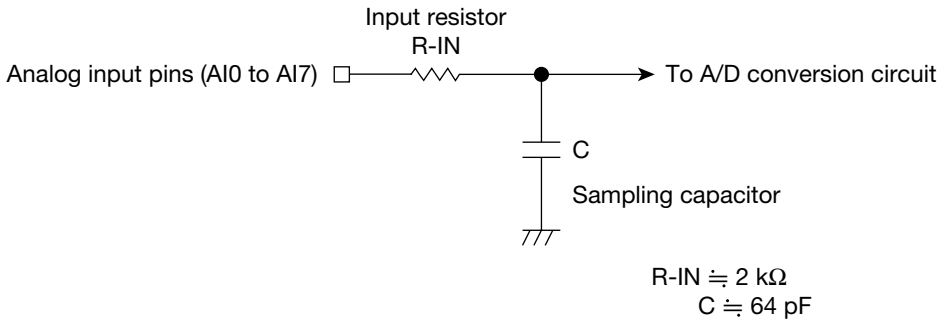


Figure 13-8 Equivalent Circuit of Analog Input Section

Table 13-2 Clock Allocation in A/D Conversion Processes

ADTM0			Number of clocks for A/D	Number of clocks required by each process		
2	1	0	conversion of 1 channel	Sampling	A/D conversion	Other
0	0	0	1536	935	314	287
0	0	1	1024	623	210	191
0	1	0	768	467	158	143
0	1	1	512	311	106	95
1	0	0	384	233	80	71
1	0	1	256	155	54	47
1	1	0	192	116	41	35
1	1	1	128	77	28	23

Units: CPUCLK

The following factors affect the conversion precision of the A/D converter.

1) Signal source impedance of the analog input → depends upon external circuit

2) Sampling time → depends upon ADTM00, ADTM01 settings

3) Actual precision of the A/D converter (comparator, CR precision, etc.)

The overall precision of the A/D converter is determined by the precision during sampling (items 1 and 2 above) and the actual precision of the A/D converter.

In consideration of the precision during sampling (dependent upon the signal source impedance), it is desirable to set a long sampling time. If the sampling time is short, it is difficult to maintain precision. In practical applications, set the conversion clock (sampling time) and design external circuitry that will satisfy the optimum requirements for "conversion time" and "conversion precision".

13.5.2 Noise-Suppression Measures

Based on the voltage difference between the analog reference voltage (V_{REF}) pin and the analog ground (AGND), the A/D converter in the MSM66577 family converts an analog voltage at the analog input pin into digital data. Because this type of A/D converter does not have a reference voltage source inside the microcomputer, "stability" and "noise-suppression measures" for V_{REF} and AGND are important.

As noise-suppression measures, insert a bypass capacitor between the analog reference voltage (V_{REF}) pin and the analog ground (AGND) pin. Also, connect the analog ground (AGND) to a stable GND on the circuit board.

If the digital and analog layouts can be separated on the circuit board, separate the circuit into a digital system (V_{DD}/GND) and an analog system (V_{REF}/AGND). Connect bypass capacitors to each system to reduce the circulation of GND noise in the digital system. Divide the circuit board into separate GND planes for the digital and analog systems, and then connect each GND plane to a common location where there is a stable GND supply.

In addition to inserting a bypass capacitor of 10 µF to 47 µF or larger between V_{REF} and AGND, the stability of V_{REF} can be maintained by connecting a 0.01 µF to 0.1 µF high-pass capacitor in parallel. Because the V_{REF} voltage supply is used to avoid the effect of digital noise on the comparator used in A/D conversion, adding a high-pass capacitor is effective in reducing V_{REF} fluctuations.

Figure 13-9 shows an example of noise-suppression measures.

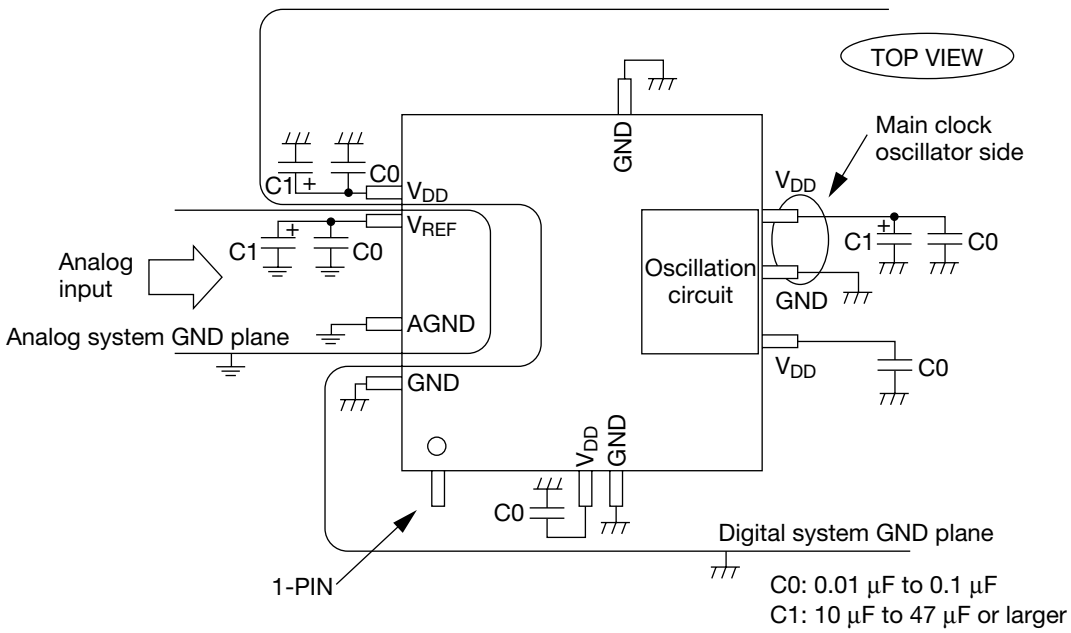


Figure 13-9 Example of Noise-Suppression Measures

13.6 A/D Converter Interrupt

When each of A/D converter interrupt factors occurs, the interrupt request flag (QAD) is set to "1". The interrupt request flag (QAD) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (EAD). The interrupt enable flag (EAD) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0AD and P1AD). The interrupt priority setting flags (P0AD and P1AD) are located in interrupt priority control register 7 (IP7).

Table 13-3 lists the vector address of the A/D converter interrupt factors and the interrupt processing flags.

Table 13-3 A/D Converter Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
A/D conversion of one cycle of the scan channels is complete	0044	QAD	EAD	P1AD	P0AD
A/D conversion of the select mode is complete					
Symbols (byte) of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	17-15	17-20	17-29	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 14

D/A Converter Functions

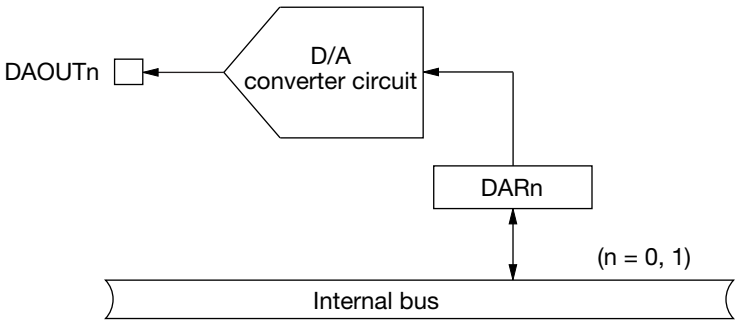
14. D/A Converter Functions

14.1 Overview

The MSM66577 family has an internal 2-channel 8-bit D/A converter. The D/A conversion method utilizes ladder resistors. If values desired to be output are written to 8-bit DA registers (DAR0, DAR1), the converted analog levels will be output from the port secondary function output pins.

14.2 D/A Converter Configuration

Figure 14-1 shows the D/A converter configuration.



DAR0: DA register 0
DAR1: DA register 1
DACON: DA control register
DAOUT0: D/A conversion output 0 (P14_6)
DAOUT1: D/A conversion output 1 (P14_7)

Figure 14-1 8-Bit D/A Converter Configuration

14.3 D/A Converter Registers

Table 14-1 lists a summary of SFRs for control of the D/A converter.

Table 14-1 Summary of SFRs for D/A Converter Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 operation	Initial value [H]	Reference page
00DD	D/A control register	DACON	—	R/W	8	FE	14-2
00DE	DA register 0	DAR0	—	R/W	8	00	14-2
00DF	DA register 1	DAR1	—	R/W	8	00	14-2

14.3.1 Description of D/A Converter Registers

(1) DA registers (DAR0, DAR1)

DA registers (DAR0, DAR1) consist of 8 bits. After configuring the corresponding port as a secondary function output and selecting AOn, if the desired output value is written to a DA register, the converted analog level will be output from the corresponding port.

If a value other than 00H is written to a DA register, the DA section will consume power to perform the conversion. Therefore, when not using D/A converters, write 00H to the DA registers.

DA registers can be read from and written to by the program,.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), DAR0 and DAR1 become 00H.

(2) DA control register (DACON)

The DA control register (DACON) consists of 1 bit. Bit 0 (DAON) of DACON specifies whether D/A conversion operates or is halted during standby.

If the STOP mode is transferred to while DAON is "0", even if the port secondary function has been set and AOn selected, D/A conversion output will be discontinued while stopped, and current that is consumed at the D/A section will be cut. At that time, DA register contents will be preserved.

If DAON is set to "1" and the STOP mode transferred to, AOn will continue to be output.

DACON can be read from and written to by the program. However, write operations are invalid for bits 1 to 7. If read, bits 1 to 7 are always "1".

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), DACON becomes FEH.

Figure 14-2 shows the DACON configuration.

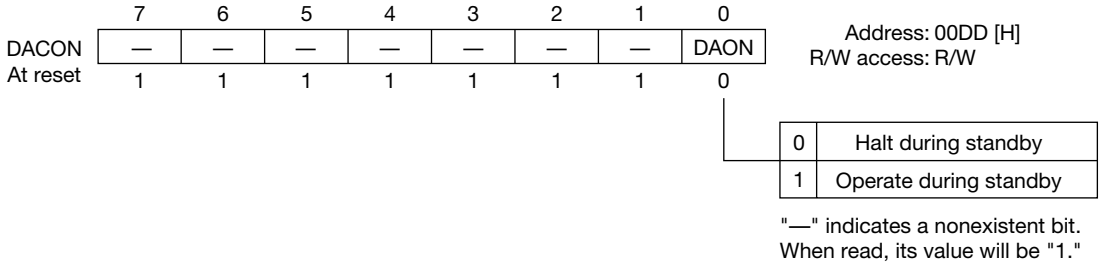


Figure 14-2 DACON Configuration

14.3.2 Example of D/A Converter-related Register Settings

(1) Port 14 mode register (P14IO)
If D/A conversion output is to be used, set bits 6 and 7 (P14IO6, P14IO7) to "1" to configure the ports as outputs.

(2) Port 14 secondary function control register (P14SF)
If D/A conversion output is to be used, set bits 6 and 7 (P14SF6, P14SF7) to "1" to configure the ports as secondary function outputs.

(3) DA registers (DAR0, DAR1)
Write a digital value to DAR0 that is equivalent to the analog level desired to be output from pin AO0.

Write a digital value to DAR1 that is equivalent to the analog level desired to be output from pin AO1.

(4) DA control register (DACON)
With bit 0 (DAON), specify whether D/A conversion will operate or be halted during standby.

14.3.3 D/A Converter Operation

The 8-bit DA registers (DAR0, DAR1) are initialized to 00H. While in this state, if ports 14_6 and 14_7 (DAOUT0, DAOUT1) are configured as secondary function outputs, the output will be at GND level (there is no current path in this state). At this time, if the port data registers are read, a value of "1" will be obtained regardless of the D/A output level. If the desired output value is written to DA registers (DAR0, DAR1), the converted analog level will be output from the corresponding port. Since the analog output is not buffered inside the chip, current cannot be drawn out. Therefore, connect a buffering amp if necessary.

Bit 0 (DAON) of the DA control register (DACON) is initialized to 0. At this time, if bit 2 (FLT) of SBCON is set to "1" and the STOP mode entered, ports 14_6 and 14_7 (DAOUT0, DAOUT1) will float, the D/A converter will halt, and the current path will be cut. DA register (DAR0, DAR1) contents will be preserved.

With bit 0 (DAON) of the DA control register (DACON) set to "1", if the STOP mode is entered, the D/A converter will not halt, and analog levels will continue to be output from ports 14_6 and 14_7 (DAOUT0, DAOUT1).

If a value n is written to DA registers (DAR0, DAR1), the analog level obtained is expressed as the following.

$$V_{DD} \times n/256 \text{ [V]}$$

When V_{DD} is 5 V, even if 0FFH is written to DA registers (DAR0, DAR1), the analog output will be approximately 4.98 V. A 5 V full-scale result cannot be obtained. Therefore, if a 5 V full-scale analog level is necessary, reconfigure the port as a primary function output.

Chapter 15

Peripheral Functions

15. Peripheral Functions

15.1 Overview

The MSM66577 family has the following functions to service peripheral ICs: a clock out function, an external XTCLK input control function, a HOLD input control function, and a WAIT input control function. These functions can be specified with the peripheral control register (PRPHCON).

15.2 Description of Each Peripheral Function

15.2.1 Clock Out Function

The clock out function has following two functions :

- To output a frequency divided clock of the main clock (OSCCLK) via the CLKOUT pin.
- To output the subclock (XTCLK) via the XTOUT pin.

The main clock frequency division ratio is specified with bit 0 and bit 1 (CLKO0 and CLKO1) of the peripheral control register (PRPHCON).

When the CLKOUT pin is to be used, P11_2 must be configured as a secondary function output.

When the XTOUT pin is used to output the subclock (XTCLK), P11_3 must be configured as a secondary function output.

15.2.2 External XTCLK Input Control Function

Because XT oscillation operates on an internally regulated voltage, an external CLK cannot normally be input to the oscillation pin. However, if bit 4 (EXTXT) of the peripheral control register (PRPHCON) is set to "1", the internally regulated voltage is switched to V_{DD} and the oscillation feedback resistor is turned off, enabling the input of an external XTCLK (V_{DD} level) to the XT pin.

15.2.3 HOLD Input Control Function

If the HOLD mode, a standby function, is to be used, set bit 5 (HOLD) of the peripheral control register (PRPHCON) to "1". Configuring P9_7 as a secondary function output (HLDACK) enables the output of a signal that indicates availability of the bus (to transfer to the HOLD mode).

15.2.4 WAIT Input Control Function

Setting bit 6 (WAIT) of the peripheral control register (PRPHCON) to "1" enables wait cycles to be inserted by an external device when accessing an external data memory area.

15.3 Peripheral Control Register (PRPHCON)

The peripheral control register (PRPHCON) consists of 5 bits.

Bits 0 and 1 (CLKO0 and CLKO1) specify the frequency division ratio of OSCCLK that is output from the CLKOUT pin. If bit 4 (EXTXT) is set to "1", an external clock can be input to the XT oscillation circuit. Bit 5 (HOLD) enables or disables the HOLD pin input. Bit 6 (WAIT) enables or disables the WAIT pin input.

PRPHCON can be read from and written to by the program. However, write operations are invalid for bits 2, 3 and 7. If read, a value of "1" will always be obtained for bits 2, 3 and 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PRPHCON becomes 8CH.

Figure 15-1 shows the PRPHCON configuration.

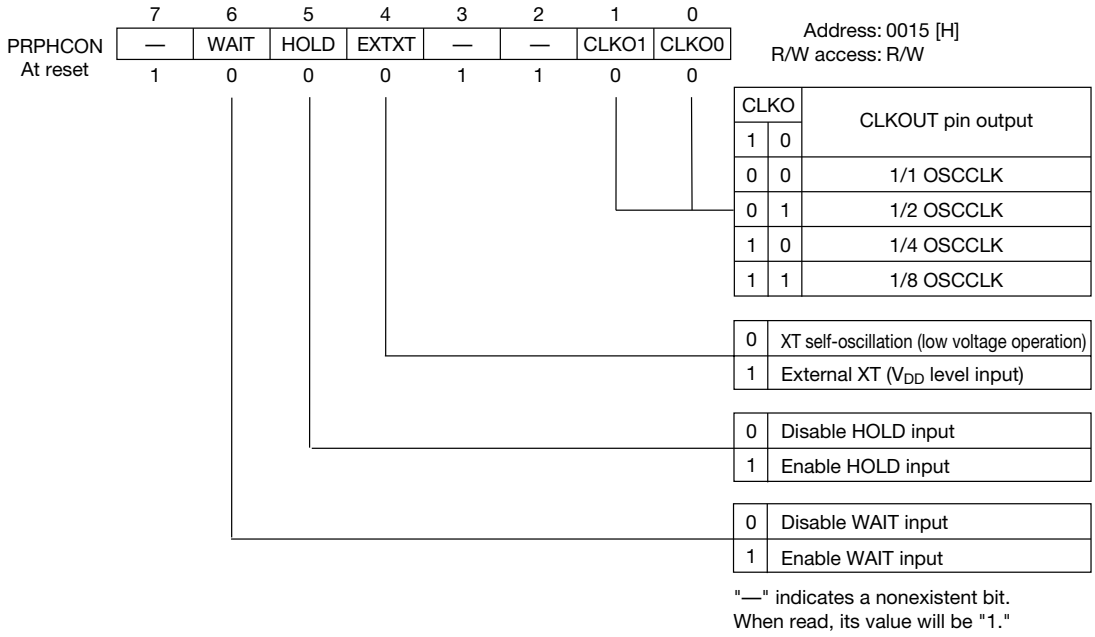


Figure 15-1 PRPHCON Configuration

Chapter 16

External Interrupt Functions

16. External Interrupt Functions

16.1 Overview

The MSM66577 family is equipped with 9 external interrupt inputs that can be classified into 2 categories. One category is maskable interrupts, of which there are 8 (EXINT0 to EXINT7). The other category is non-maskable interrupts, and there is 1 (NMI).

EXINT0 to EXINT7 are assigned as secondary functions of ports P6_0 to P6_3 and P9_0 to P9_3. If EXINT are to be used, configure the corresponding ports as inputs.

NMI has its own dedicated pin.

16.2 External Interrupt Registers

Table 16-1 lists a summary of SFRs for the control of external interrupts.

Table 16-1 Summary of SFRs for External Interrupt Control

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0058	External Interrupt Control Register 0	EXI0CON	—	R/W	8	00	16-2
0059☆	External Interrupt Control Register 1	EXI1CON	—	R/W	8	00	16-3
005A☆	External Interrupt Control Register 2	EXI2CON	—	R/W	8	0C/4C	16-4

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

16.2.1 Description of External Interrupt Registers

(1) External interrupt control register 0 (EXI0CON)

The external interrupt control register 0 (EXI0CON) consists of 8 bits and sets external interrupts EXINT0 to EXINT3. For each external interrupt setting, EXI0CON specifies the valid edge (falling edge, rising edge, or both edges) or the interrupt input invalid.

EXI0CON can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), EXI0CON becomes 00H.

Figure 16-1 shows the configuration of EXI0CON.

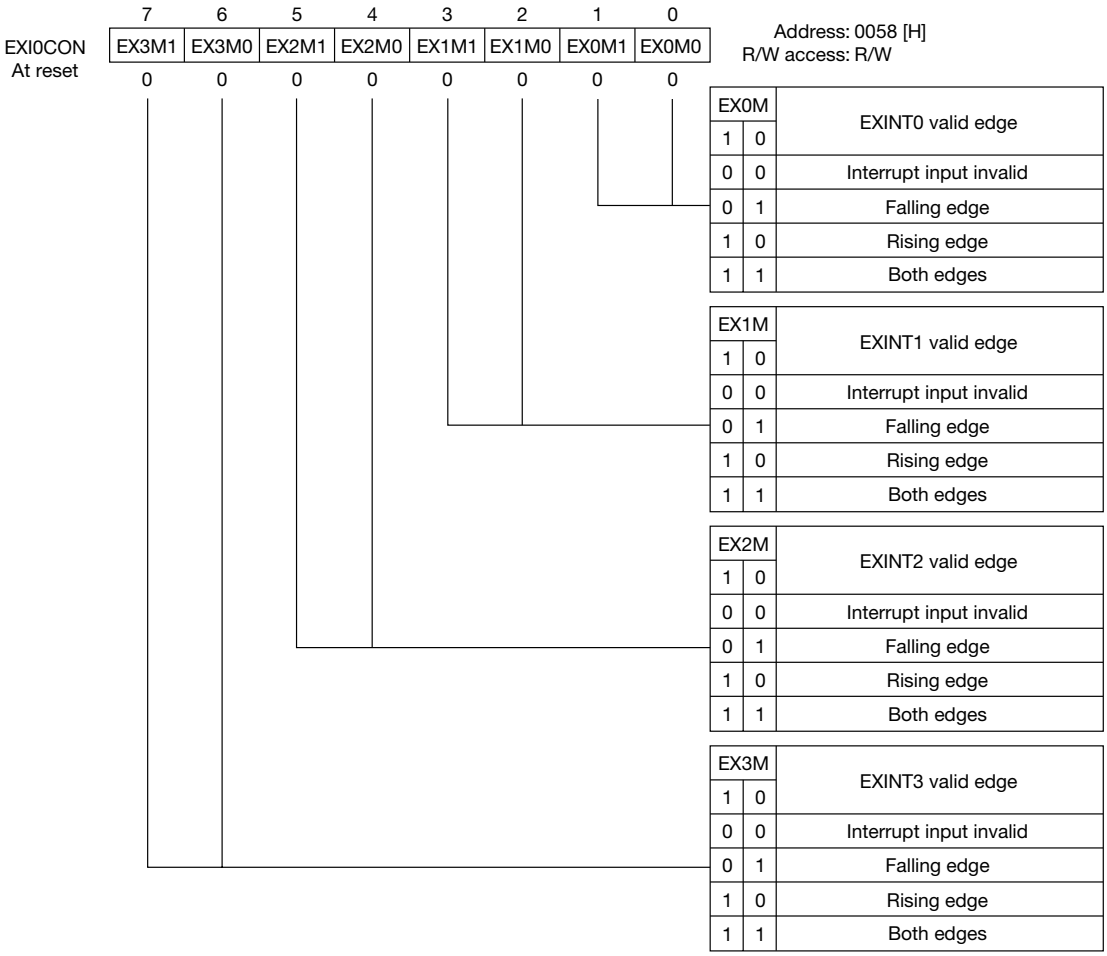


Figure 16-1 EXI0CON Configuration

(2) External interrupt control register 1 (EXI1CON)

The external interrupt control register 1 (EXI1CON) consists of 8 bits and sets external interrupts EXINT4 to EXINT7. For each external interrupt setting, EXI1CON specifies the valid edge (falling edge, rising edge, or both edges) or the interrupt input invalid.

EXI1CON can be read from and written to by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), EXI1CON becomes 00H.

Figure 16-2 shows the configuration of EXI1CON.

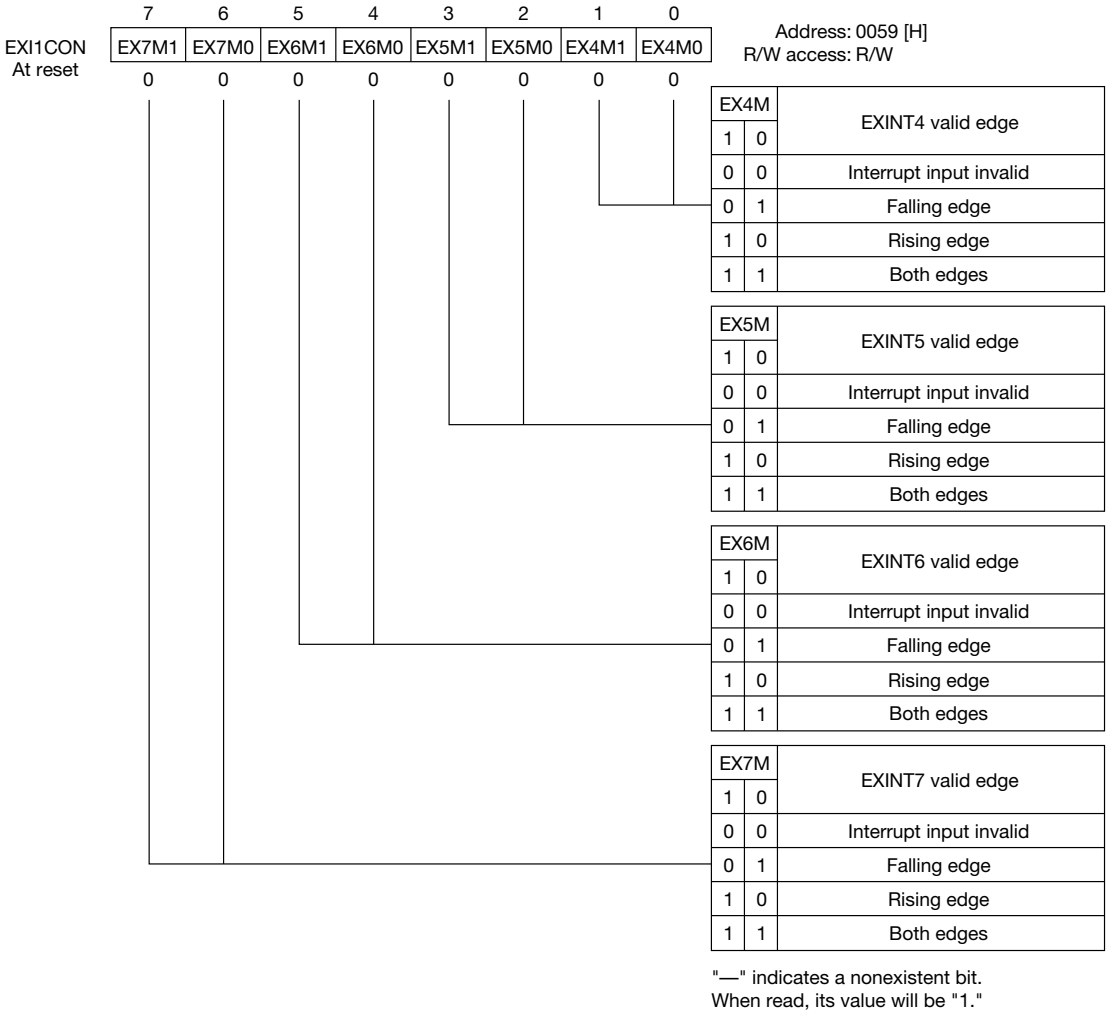


Figure 16-2 EXI1CON Configuration

(3) External interrupt control register 2 (EXI2CON)

The external interrupt control register 2 (EXI2CON) consists of 4 bits. Bits 4 and 5 (NMIM0 and NMIM1) specify the valid edge for NMI. Bit 7 (MIPF) enables or disables priority control for all maskable interrupts. Bit 6 (NMIRD) monitors the NMI pin.

EXI2CON can be read from and written to by the program. However, write operations to the lower 4 bits and bit 6 are invalid. If read, bits 0 and 1 will always be "0", and bits 2 and 3 will be "1". The NMI pin level is read from bit 6 (NMIRD). This bit can be conveniently used by the program to read the pin level during a NMI routine.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), EXI2CON becomes 0CH if the NMI pin is at a low level, or 4CH if the NMI pin is at a high level.

Figure 16-3 shows the configuration of EXI2CON.

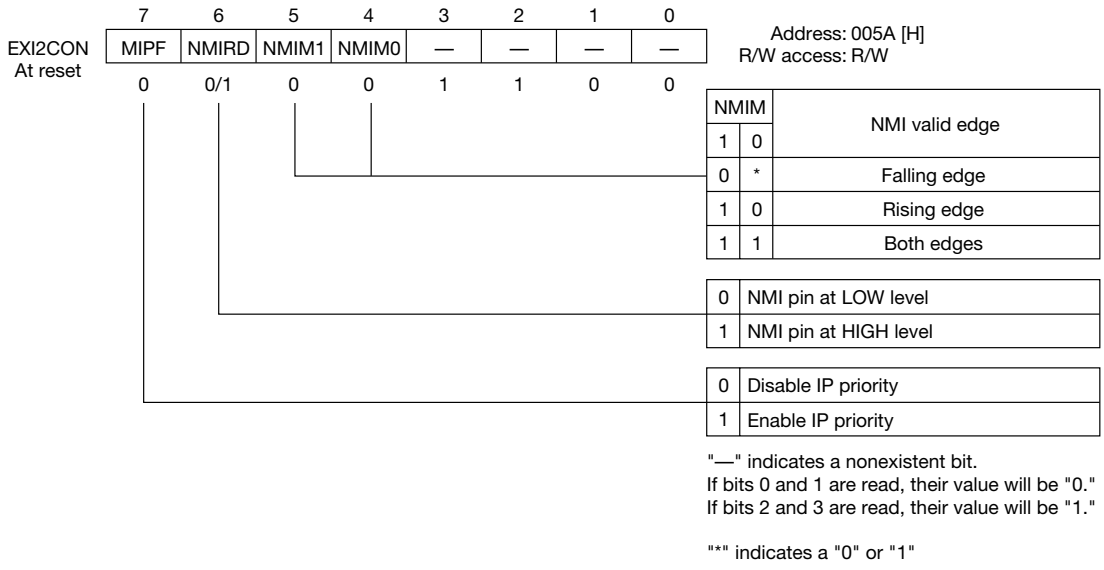


Figure 16-3 EXI2CON Configuration

16.2.2 Example of External Interrupt-related Register Settings

(1) Port 6 mode register (P6IO)
If EXINT0 to EXINT3 are to be used, reset the corresponding bits 0 to 3 (P6IO0 to P6IO3) to "0" to configure those ports as inputs.

(2) Port 9 mode register (P9IO)
If EXINT4 and/or EXINT5 are to be used, reset the corresponding bits 0 to 3 (P9IO0 to P9IO3) to "0" to configure those ports as inputs.

(3) Port 6 secondary function control register (P6SF)
If EXINT0 to EXINT3 are to be used, enable or disable pull-up resistors with the corresponding bits 0 to 3 (P6SF0 to P6SF3).

(4) Port 9 secondary function control register (P9SF)
If EXINT4 and/or EXINT5 are to be used, enable or disable pull-up resistors with the corresponding bits 0 to 3 (P9SF0 to P9SF3).

(5) External interrupt control register 0 (EXI0CON)
If EXINT0 is to be used, specify the valid edge with bits 0 and 1 (EX0M0, EX0M1). If EXINT1, EXINT2 and/or EXINT3 are to be used, specify a valid edge for each with bits 2 and 3 (EX1M0, EX1M1), bits 4 and 5 (EX2M0, EX2M1), and bits 6 and 7 (EX3M0, EX3M1).

(6) External interrupt control register 1 (EXI1CON)
If EXINT4 is to be used, specify the valid edge with bits 0 and 1 (EX4M0, EX4M1). If EXINT5, EXINT6 and EXINT7 are to be used, specify the valid edge with bits 2 and 3 (EX5M0, EX5M1), bits 4 and 5 (EX6M0, EX6M1), and bits 6 and 7 (EX7M0, EX7M1).

(7) External interrupt control register 2 (EXI2CON)
Specify the NMI valid edge with bits 4 and 5 (NMIM0, NMIM1). If interrupt priority is to be used, set bit 7 (MIPF) to "1".

16.3 EXINT0 to EXINT7 Interrupts

When a valid edge is input to each external interrupt input pin, the corresponding interrupt request flag is set to "1". The interrupt request flags are located in interrupt request registers 0 to 2 (IRQ0 to IRQ2).

Interrupts can be enabled or disabled by the interrupt enable flag that corresponds to each pin input. The interrupt enable flags are located in interrupt enable registers 0 to 2 (IE0 to IE2).

Three levels of priority can be set with the interrupt priority setting flags that correspond to each pin input. The interrupt priority setting flags are located in interrupt priority control registers 0, 2 and 4 (IP0, IP2 and IP4).

Table 16-2 lists the vector addresses for each pin input of EXINT0 to EXINT7 and the interrupt processing flags.

*n (n = 1 to 9) in the above table indicates the register in which each flag is allocated.

Table 16-2 EXINT0 to EXINT7 Vector Addresses and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level 1	Priority level 0
EXINT0 pin input (external interrupt 0)	000A	QINT0 *1	EINT0 *4	P1INT0	P0INT0 *7
EXINT1 pin input (external interrupt 1)	001C	QINT1 *2	EINT1 *5	P1INT1	P0INT1 *8
EXINT2 pin input (external interrupt 2)	001E	QINT2	EINT2	P1INT2	P0INT2
EXINT3 pin input (external interrupt 3)	0020	QINT3	EINT3	P1INT3	P0INT3
EXINT4 pin input (external interrupt 4)	002A	QINT4 *3	EINT4 *6	P1INT4	P0INT4 *9
EXINT5 pin input (external interrupt 5)	002C	QINT5	EINT5	P1INT5	P0INT5
EXINT6 pin input (external interrupt 6)	002E	QINT6	EINT6	P1INT6	P0INT6
EXINT7 pin input (external interrupt 7)	0030	QINT7	EINT7	P1INT7	P0INT7
Symbols (byte) of registers that contain interrupt processing flags		IRQ0 *1 IRQ1 *2 IRQ2 *3	IE0 *4 IE1 *5 IE2 *6	IP0 *7 IP2 *8 IP4 *9	
	Reference page	17-12 *1 17-13 *2 17-14 *3	17-17 *4 17-18 *5 17-19 *6	17-22 *7 17-24 *8 17-26 *9	

For further details regarding interrupt processing, refer to Chapter 17, "Interrupt Processing Functions".

Chapter 17

Interrupt Processing Functions

17. Interrupt Processing Functions

17.1 Overview

The MSM66577 family has 39 types of interrupts (9 external and 30 internal). These are assigned to 30 vectors. One of the external interrupts is a non-maskable interrupt. Three levels of priority can be set for maskable interrupts.

Table 17-1 lists interrupts and their corresponding vector addresses.

Table 17-1 Interrupts and Their Corresponding Vector Addresses

Interrupt	Vector address [H]
NMI pin input (non-maskable interrupt)	0008
EXINT0 pin input (external interrupt 0)	000A
Free running counter overflow	000C
CPCM0 event input, compare match	0016
CPCM1 event input, compare match	0018
Timer 0 overflow	001A
EXINT1 pin input (external interrupt 1)	001C
EXINT2 pin input (external interrupt 2)	001E
EXINT3 pin input (external interrupt 3)	0020
Timer 1 overflow	0022
Timer 2 overflow	0024
Timer 3 overflow	0026
EXINT4 pin input (external interrupt 4)	002A
EXINT5 pin input (external interrupt 5)	002C
EXINT6 pin input (external interrupt 6)	002E
EXINT7 pin input (external interrupt 7)	0030
Timer 4 overflow	0036
SIO1 transmit buffer empty, transmit complete, receive complete	0038
Timer 5 overflow	003A
Interrupt by SIO5 transfer completion	003C
Interrupt by SIO6 transmit buffer empty, transmit completion, receive completion	003E
Interrupt by SIO4 transfer completion	0040
Timer 6 overflow	0042
One cycle of A/D conversion scan channels complete, A/D conversion select mode complete	0044
Real-time counter output (interval: 0.125 to 1 s)	0048
PWC0 overflow, match of PWC0 and PWR0	006A
PWC1 overflow, match of PWC1 and PWR1	006C
Match of PWC0 and PWR2	006E
Match of PWC1 and PWR3	0070
Timer 9 overflow	0072

17.2 Interrupt Function Registers

Table 17-2 lists a summary of SFRs for interrupt processing

Table 17-2 Summary of SFRs for Interrupt Processing

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
0004	Program status word	PSWL	PSW	R/W	8/16	00	2-18
0005		PSWH				00	
005A☆	External interrupt control register 2	EXI2CON	—	R/W	8	0C/4C	16-4
0030☆	Interrupt request register 0	IRQ0	—	R/W	8	00	17-12
0031☆	Interrupt request register 1	IRQ1	—	R/W	8	00	17-13
0032☆	Interrupt request register 2	IRQ2	—	R/W	8	00	17-14
0033☆	Interrupt request register 3	IRQ3	—	R/W	8	00	17-15
005C☆	Interrupt request register 4	IRQ4	—	R/W	8	E0	17-16
0034☆	Interrupt enable register 0	IE0	—	R/W	8	00	17-17
0035☆	Interrupt enable register 1	IE1	—	R/W	8	00	17-18
0036☆	Interrupt enable register 2	IE2	—	R/W	8	00	17-19
0037☆	Interrupt enable register 3	IE3	—	R/W	8	00	17-20
005D☆	Interrupt enable register 4	IE4	—	R/W	8	E0	17-21
0038☆	Interrupt priority control register 0	IP0	—	R/W	8	00	17-22
0039☆	Interrupt priority control register 1	IP1	—	R/W	8	00	17-23
003A	Interrupt priority control register 2	IP2	—	R/W	8	00	17-24
003B☆	Interrupt priority control register 3	IP3	—	R/W	8	00	17-25
003C☆	Interrupt priority control register 4	IP4	—	R/W	8	00	17-26
003D☆	Interrupt priority control register 5	IP5	—	R/W	8	00	17-27
003E☆	Interrupt priority control register 6	IP6	—	R/W	8	00	17-28
003F☆	Interrupt priority control register 7	IP7	—	R/W	8	00	17-29
005E☆	Interrupt priority control register 8	IP8	—	R/W	8	00	17-30
005F☆	Interrupt priority control register 9	IP9	—	R/W	8	FC	17-31

[Notes]

1. Addresses may not be consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

17.3 Description of Interrupt Processing

17.3.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt (NMI) is an external interrupt that cannot be masked.

When the valid edge specified by bits 4 and 5 (NMIM0, NMIM1) of EXI2CON is detected, the CPU immediately transfers processing to the non-maskable interrupt.

However, the one exception occurs after reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), where the non-maskable interrupt is masked until execution of the first instruction is complete. This function is intended to prevent loss of program control after reset in the case where the non-maskable interrupt occurs before the system stack pointer (SSP) is set with a value (when the SSP is undefined). Therefore, operated as part of the above NMI function, set an appropriate value in SSP with the "first instruction after reset".

[Related information reference guide]
NMI settings … page 16-4

When the non-maskable interrupt (NMI) occurs, a sequence such as listed below is automatically processed by the hardware and the first instruction of the NMI routine is executed. 14 cycles are used to transfer to the NMI routine.

- Save the program counter (PC)
- Save the accumulator (ACC)
- Save the local register base (LRB)
- Save the program status word (PSW)
- Reset the non-maskable interrupt request flag
- Disable maskable interrupts
- Disable multiple interrupts by the non-maskable interrupt
- Load the program counter with the value that has been written to the NMI routine vector table (0008H, 0009H)

Use a RTI instruction at the end of the NMI routine.

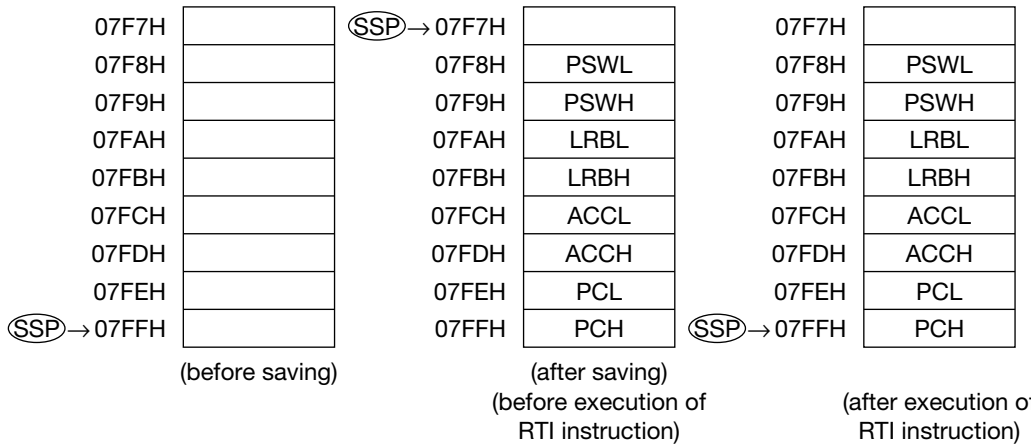
When a RTI instruction is executed, the hardware automatically processes a sequence such as listed below to complete the NMI routine. 12 cycles are used to return from the NMI routine.

- Restore the program status word (PSW)
- Restore the local register base (LRB)
- Restore the accumulator (ACC)
- Restore the program counter (PC)
- Enable maskable interrupts
- Enable multiple interrupts by the non-maskable interrupt

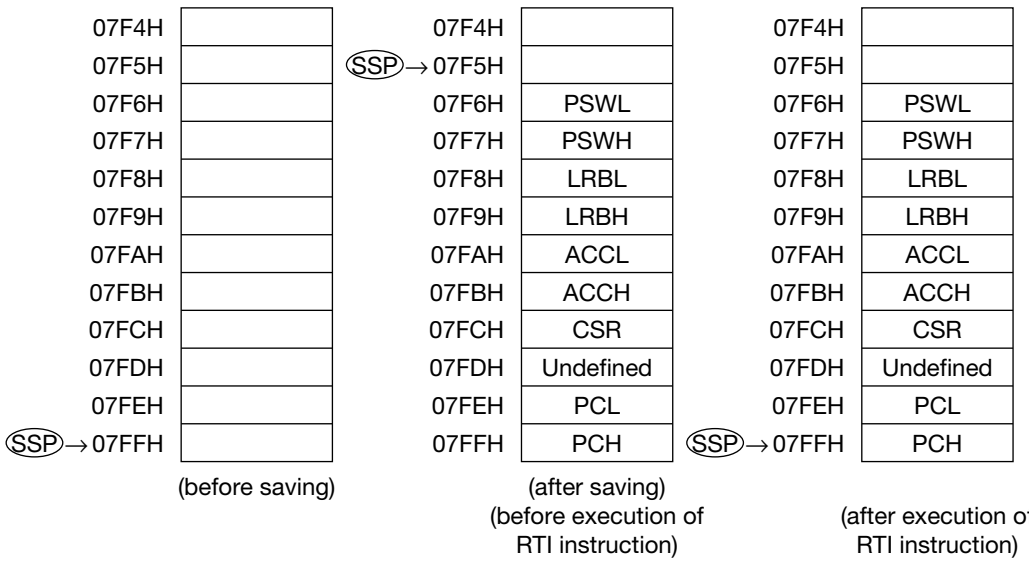
Figure 17-1 shows examples of saving and restoring the PC, ACC, LRB and PSW.

[Note]
If the program memory space has been expanded to 1MB, in addition to the above processing, the code segment register (CSR) will be saved and restored. In this case, 17 cycles will be used to transfer to the NMI routine, and 14 cycles to return from the NMI routine.

- Interrupt processing example (for a 64KB program memory space)



- Interrupt processing example (for a greater than 64KB program memory space)



SSP: System Stack Pointer

Figure 17-1 Examples of Saving and Restoring the PC, ACC, LRB and PSW

17.3.2 Maskable Interrupts

Maskable interrupts are generated by various interrupt factors such as built-in internal peripheral hardware, external interrupt inputs, etc.

The control of maskable interrupts is performed by the following.
- Interrupt request registers (IRQ0 to IRQ4)
- Interrupt enable registers (IE0 to IE4)
- Master interrupt enable flag (MIE)
- Master interrupt priority flag (MIPF)
- Interrupt priority control registers (IP0 to IP9)

(1) Interrupt request registers (IRQ0 to IRQ4)

Interrupt request registers (IRQs) are set to "1" when each interrupt source generates an interrupt signal. If an interrupt is received, the registers are automatically reset to "0" while transferring to the interrupt processing routine. IRQ bits can also be set to "1" or "0" by the program.

(2) Interrupt enable registers (IE0 to IE4)

Interrupt enable registers (IEs) individually enable or disable the generation of interrupts. When an IE bit is "0", generation of the corresponding interrupt is disabled. When an IE bit is "1", generation of the corresponding interrupt is enabled.

(3) Master interrupt enable flag (MIE)

The master interrupt enable flag (MIE) is a 1-bit flag located in the program status word (PSW). MIE enables or disables generation of all the maskable interrupts.
MIE = "0" All maskable interrupts are disabled (regardless of IE)
MIE = "1" Maskable interrupts are enabled (only those interrupt factors enabled by IE)

[Related information reference guide]
Program status word (PSW) … Page 2-18

(4) Master interrupt priority flag (MIPF)

The master interrupt priority flag (MIPF) is a 1-bit flag located in the external interrupt control register 2 (EXI2CON). MIPF enables or disables priority for all the maskable interrupts.
MIPF = "0" Priority control disabled (regardless of IP, interrupts controlled by MIE and IE only)
MIPF = "1" Priority control enabled (3 levels of priority control according to IP setting)

[Related information reference guide]
External interrupt control register 2 (EXI2CON) … Page 16-4

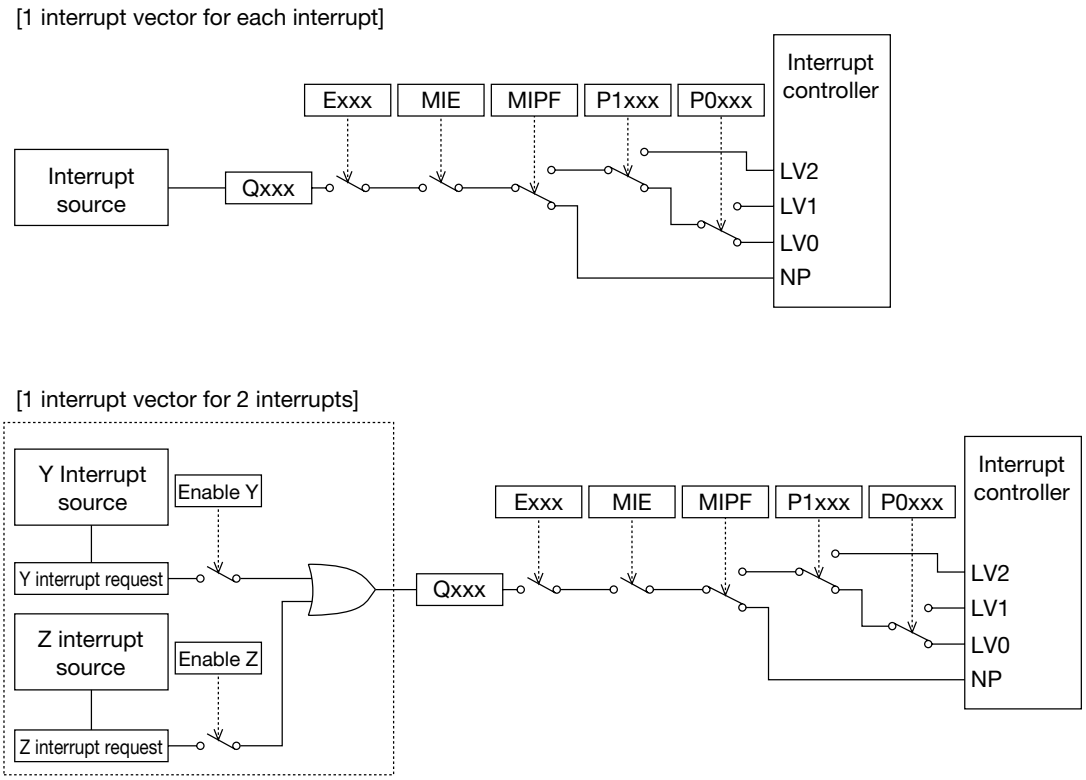
(5) Interrupt priority control registers (IP0 to IP9)

Interrupt priority control registers (IPs) specify the priority of maskable interrupts. The 2-bit specification (P1xxx, P0xxx) for each interrupt indicates 3 levels of priority (where xxx is an abbreviation for each interrupt factor). For further details regarding priority control, refer to Section 17.3.3, "Priority Control of Maskable Interrupts".
Priority is specified as shown below.

P1xxx	P0xxx	Priority
0	0	Level 0 (low)
0	1	Level 1 ↕
1	*	Level 2 (high)

(* indicates either "0" or "1")

Figure 17-2 shows a block diagram of the control for maskable interrupts. IRQ bits are indicated as Qxxx, IE bits as Exxx, and IP bits as P0xxx, P1xxx for each interrupt factor. In some cases, several maskable interrupts correspond to the same interrupt vector. For those interrupts, within each function block there is a flag to enable or disable multiple interrupts and an interrupt request flag to verify (by polling) which interrupt was generated.



The control in the above enclosed area exists in each function block.

Figure 17-2 Maskable Interrupt Control Block Diagram

Table 17-3 lists the vector address and bit symbol for each maskable interrupt.
If multiple maskable interrupts are generated simultaneously, the lower vector address (in the order of Table 17-3) is given priority and processed. Similarly, for interrupts that have been enabled, if the priority level is set and priority control enabled (MIPF = "1"), when multiple maskable interrupts with the same priority are generated simultaneously, the lower vector address is given priority and processed.

Table 17-3 Vector Addresses and Bit Symbols for Maskable Interrupts

No.	Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
					1	0
1	EXINT0 pin input (external interrupt 0)	000A	QINT0	EINT0	P1INT0	P0INT0
2	Free running counter overflow	000C	QFRCOV	EFRCOV	P1FRCOV	P0FRCOV
3	CPCM0 event input, compare match	0016	QCPCM0	ECPCM0	P1CPCM0	P0CPCM0
4	CPCM1 event input, compare match	0018	QCPCM1	ECPCM1	P1CPCM1	P0CPCM1
5	Timer 0 overflow	001A	QTM0OV	ETM0OV	P1TM0OV	P0TM0OV
6	EXINT1 pin input (external interrupt 1)	001C	QINT1	EINT1	P1INT1	P0INT1
7	EXINT2 pin input (external interrupt 2)	001E	QINT2	EINT2	P1INT2	P0INT2
8	EXINT3 pin input (external interrupt 3)	0020	QINT3	EINT3	P1INT3	P0INT3
9	Timer 1 overflow	0022	QTM1OV	ETM1OV	P1TM1OV	P0TM1OV
10	Timer 2 overflow	0024	QTM2OV	ETM2OV	P1TM2OV	P0TM2OV
11	Timer 3 overflow	0026	QTM3OV	ETM3OV	P1TM3OV	P0TM3OV
12	EXINT4 pin input (external interrupt 4)	002A	QINT4	EINT4	P1INT4	P0INT4
13	EXINT5 pin input (external interrupt 5)	002C	QINT5	EINT5	P1INT5	P0INT5
14	EXINT6 pin input (external interrupt 6)	002E	QINT6	EINT6	P1INT6	P0INT6
15	EXINT7 pin input (external interrupt 7)	0030	QINT7	EINT7	P1INT7	P0INT7
16	Timer 4 overflow	0036	QTM4OV	ETM4OV	P1TM4OV	P0TM4OV
17	SIO1 transmit buffer empty, transmit complete, receive complete	0038	QSIO1	ESIO1	P1SIO1	P0SIO1
18	Timer 5 overflow	003A	QTM5OV	ETM5OV	P1TM5OV	P0TM5OV
19	Interrupt by SIO5 transfer completion	003C	QSIO5	ESIO5	P1SIO5	P0SIO5
20	Interrupt by SIO6 transmit buffer empty, transmit completion, receive completion	003E	QSIO6	ESIO6	P1SIO6	P0SIO6
21	Interrupt by SIO4 transfer completion	0040	QSIO4	ESIO4	P1SIO4	P0SIO4
22	Timer 6 overflow	0042	QTM6OV	ETM6OV	P1TM6OV	P0TM6OV
23	One cycle of A/D conversion scan channels complete, A/D conversion select mode complete	0044	QAD	EAD	P1AD	P0AD
24	Real-time counter output (interval: 0.125 to 1 s)	0048	QRTC	ERTC	P1RTC	P0RTC
25	PWC0 overflow, match of PWC0 and PWR0	006A	QPWM0	EPWM0	P1PWM0	P0PWM0
26	PWC1 overflow, match of PWC1 and PWR1	006C	QPWM1	EPWM1	P1PWM1	P0PWM1
27	Match of PWC0 and PWR2	006E	QPWM2	EPWM2	P1PWM2	P0PWM2
28	Match of PWC1 and PWR3	0070	QPWM3	EPWM3	P1PWM3	P0PWM3
29	Timer 9 overflow	0072	QTM9OV	ETM9OV	P1TM9OV	P0TM9OV

When a maskable interrupt occurs, a sequence such as listed below is automatically processed by the hardware and the first instruction of the maskable interrupt routine is executed. 14 cycles are used to transfer to the maskable interrupt routine.

- Save the program counter (PC)
- Save the accumulator (ACC)
- Save the local register base (LRB)
- Save the program status word (PSW)
- Reset the IRQ that initiated the maskable interrupt process
- Reset MIE in PSW (resetting MIE to "0" disables reception of all maskable interrupts)
- Disable reception of interrupts with the same or lower interrupt priority level (if MIPF = 1)
- Load the program counter with the value that has been written to the vector table

Use a RTI instruction at the end of the maskable interrupt routine.

When a RTI instruction is executed, the hardware automatically processes a sequence such as listed below to complete the maskable interrupt routine. 12 cycles are used to return from the maskable interrupt routine.

- Enable reception of interrupts with the same or lower interrupt priority level (if MIPF = 1)
- Restore the program status word (PSW) (set MIE to "1")
- Restore the local register base (LRB)
- Restore the accumulator (ACC)
- Restore the program counter (PC)

Figure 17-1 shows examples of saving and storing the PC, ACC, LRB and PSW.

[Note]
If the program memory space has been expanded to 1MB, in addition to the above processing, the code segment register (CSR) will be saved and restored. In this case, 17 cycles will be used to transfer to the maskable interrupt routine, and 14 cycles to return from the maskable interrupt routine.

17.3.3 Priority Control of Maskable Interrupts

The MSM66577 family can set 3 levels of priority for each maskable interrupt factor, resulting in easy to realize control of multiple interrupts. Priority control in actual programs is described below.

(1) Basic interrupt control

When a maskable interrupt occurs, since the reception of other maskable interrupts is automatically disabled (MIE = "0"), other interrupts (except for nonmaskable interrupts and reset processing) will not occur within the interrupt processing routine. If another maskable interrupt is generated during execution of the interrupt routine, that interrupt will wait for processing. In such a case, immediately after processing of the first interrupt is completed, processing of the interrupt that has been waiting will begin. (See Figure 17-3.) If several interrupts are awaiting processing, the interrupt vector with the lowest address will be processed first. (See Table 17-3.)

(2) Multiple interrupt control

During execution of an interrupt routine, other maskable interrupts may be enabled. This is known as "multiple interrupt control". If multiple interrupt control is set, the maskable interrupt disabling process (MIE = "0"), automatically performed by hardware when a maskable interrupt occurs, is cancelled within the maskable interrupt routine.

The following two methods exist for multiple interrupt control.

(i) Control by IE flags
(ii) Control by MIPF (Master Interrupt Priority Flag)

(i) Control by IE flags

In the interrupt processing routine, only those IE flags that correspond to the multiple interrupt factors to be enabled are set to "1". Multiple interrupts from other factors are disabled by setting their IE flags to "0".

Next, by setting the MIE flag to "1" within the interrupt processing routine, the reception of multiple interrupts for the enabled interrupt factors enabled by setting the IE flags to "1" will begin. (See Figure 17-4.)

If an interrupt occurs for which the corresponding IE flag is "0" while another interrupt is being processed, the interrupt will wait until the interrupt process being executed is completed and the program changes its IE flag to "1".

(ii) Control by MIPF (Master Interrupt Priority Flag)

In addition to the control of (i) above, by setting MIPF to "1", the priority of maskable interrupts can be controlled by the hardware. Of the enabled interrupt factors specified with IE = "1", multiple interrupts are enabled only for those interrupt factors whose priority is higher than that of the interrupt currently being processed. (If MIPF = "0", then all interrupt factors with IE specified as "1" will be enabled for multiple interrupts.)

If interrupts are generated having the same or lower priority than that of the interrupt process currently being executed, those interrupts will wait until completion of the interrupt process currently being executed. After completion of the interrupt process, if several interrupts are waiting, they will be executed in order of highest priority. However, if there are several interrupts with the same priority level, the interrupt with the lowest vector address will be processed first. (See Table 17-3.)

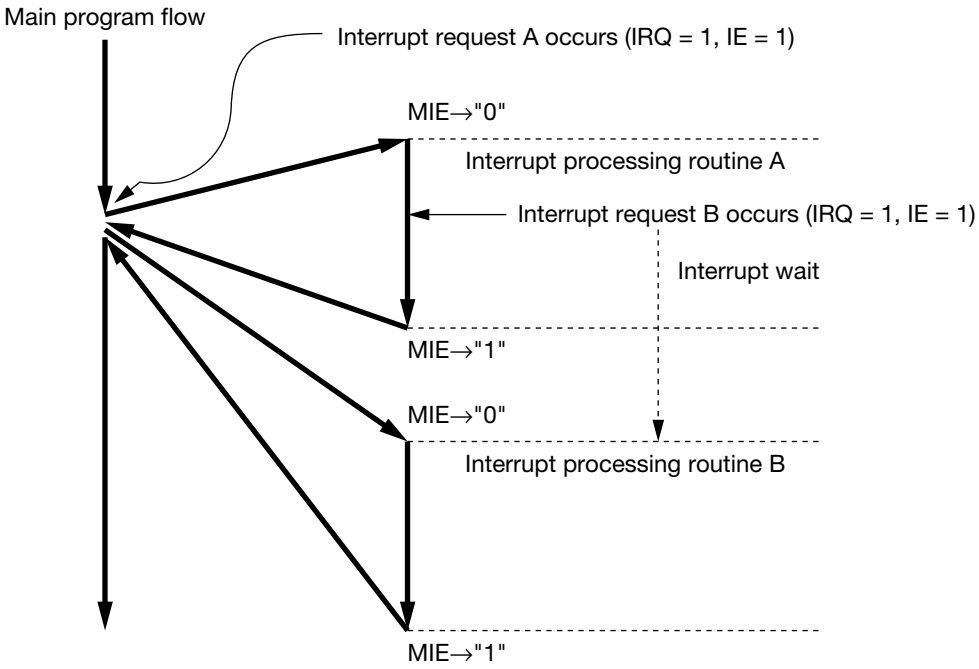


Figure 17-3 Fundamental Interrupt Control

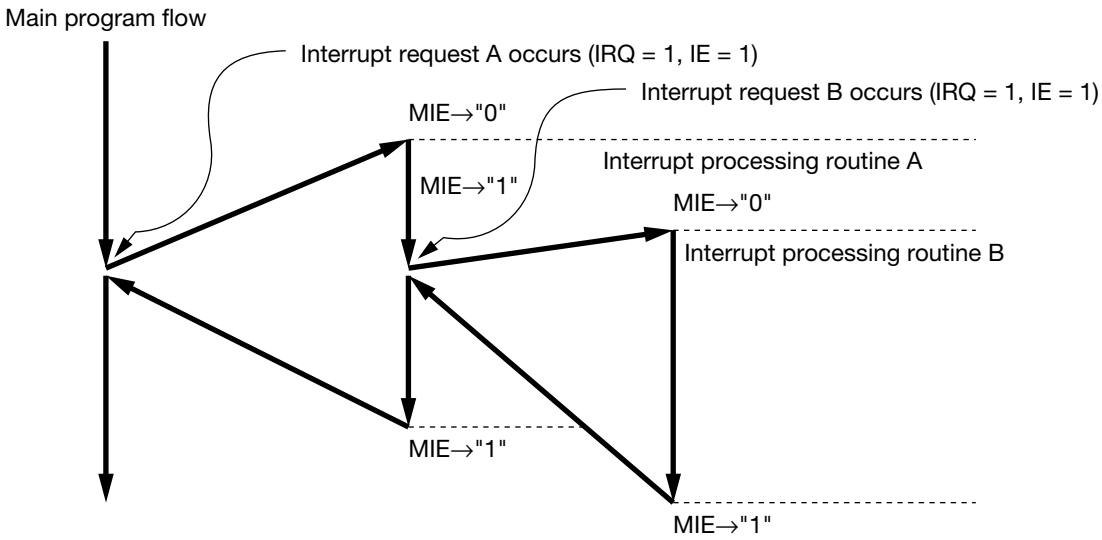


Figure 17-4 Multiple Interrupt Control

17.4 IRQ, IE and IP Register Configurations for Each Interrupt

Each interrupt factor has its own interrupt request register (IRQ0 to IRQ4), interrupt enable register (IE0 to IE4) and interrupt priority control register (IP0 to IP9).

These registers are allocated as a group of interrupt processing registers, independent from the group of operation and control registers for each internal peripheral module.

The configurations of each interrupt processing register are presented below, showing which bits of which registers are allocated as the IRQ, IE and IP flags for each interrupt factor. At the end of chapters describing internal peripheral modules, a reference page is listed for the interrupt processing registers of that module.

17.4.1 Interrupt Request Registers (IRQ0 to IRQ4)

(1) Interrupt request register 0 (IRQ0)

Interrupt request register 0 (IRQ0) consists of 4 bits. Bits are set to "1" corresponding to external interrupt 0 (bit 0), overflow of free running counter (bit 1), CPCM0 event input / compare match (bit 6), and CPCM1 event input/compare match (bit 7).

IRQ0 can be read or written by the program. However, if writing to bits 2 through 5, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 through 5.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ0 becomes 00H.

Figure 17-5 shows the configuration of IRQ0.

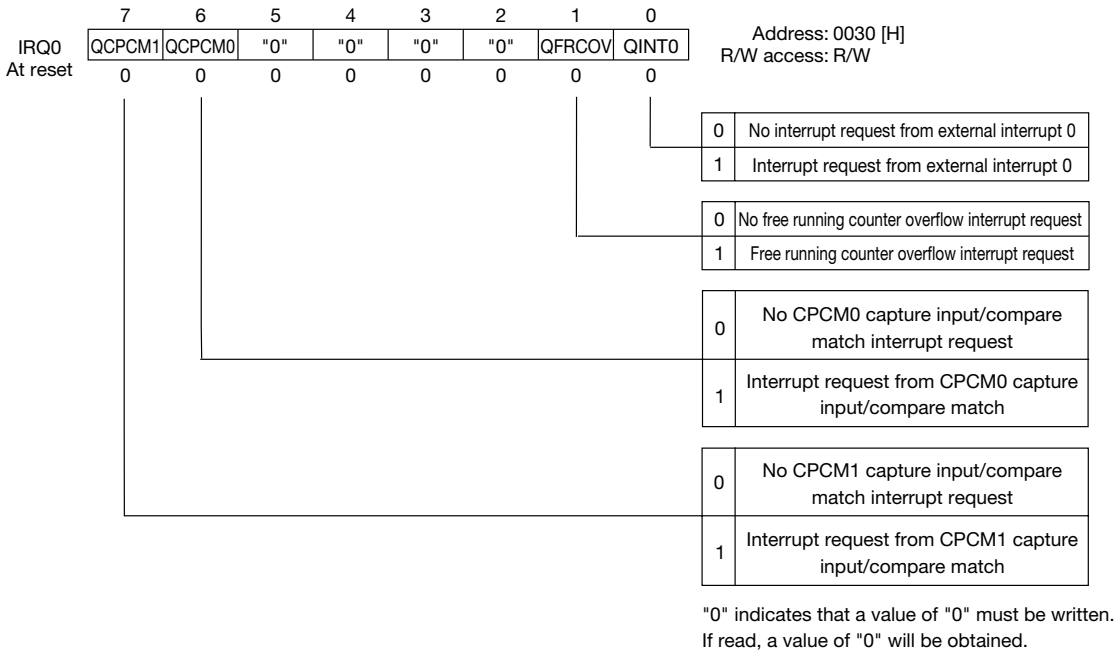


Figure 17-5 IRQ0 Configuration

(2) Interrupt request register 1 (IRQ1)

Interrupt request register 1 (IRQ1) consists of 7 bits. Bits are set to "1" corresponding to overflow of timer 0 (bit 0), external interrupts 1 to 3 (bits 1 to 3) and overflow of timers 1 to 3 (bits 4 to 6).

IRQ1 can be read or written by the program. However, if writing to bit 7, always write the bit as "0". If read, a value of "0" will always be obtained for bit 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ1 becomes 00H.

Figure 17-6 shows the configuration of IRQ1.

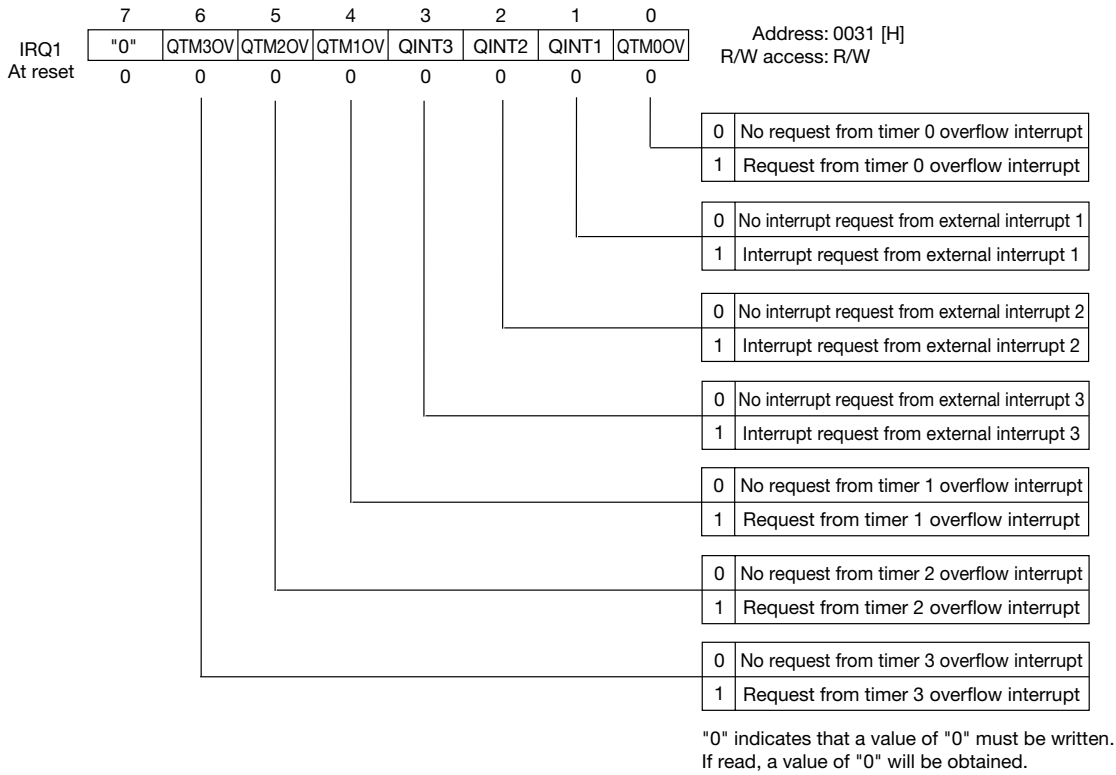


Figure 17-6 IRQ1 Configuration

(3) Interrupt request register 2 (IRQ2)

Interrupt request register 2 (IRQ2) consists of 6 bits. Bits are set to "1" corresponding to external interrupts 4 to 7 (bits 0 to 3), overflow of timer 4 (bit 6), and SIO1 transmit buffer empty/transmit complete/receive complete (bit 7).

IRQ2 can be read or written by the program. However, if writing to bits 4 and 5, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4 and 5.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ2 becomes 00H.

Figure 17-7 shows the configuration of IRQ2.

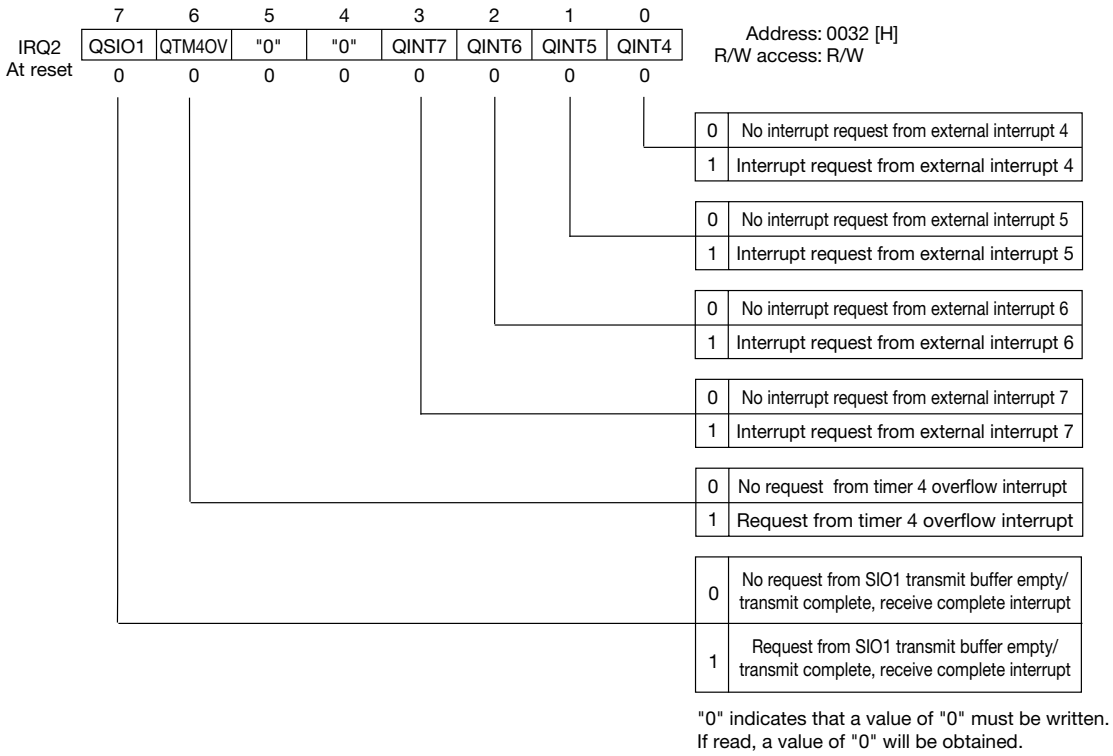


Figure 17-7 IRQ2 Configuration

(4) Interrupt request register 3 (IRQ3)

Interrupt request register 3 (IRQ3) consists of 7 bits. Bits are set to "1" corresponding to overflow of timer 5 (bit 0), SIO5 and SIO4 transmit-receive completion (bits 1 and 3), overflow of timer 6 (bit 4), A/D conversion scan channel cycle complete/select mode complete (bit 5), and real-time counter output (bit 7).

IRQ3 can be read or written by the program. However, if writing to bit 6, always write the bit as "0". If read, a value of "0" will always be obtained for bit 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ3 becomes 00H.

Figure 17-8 shows the configuration of IRQ3.

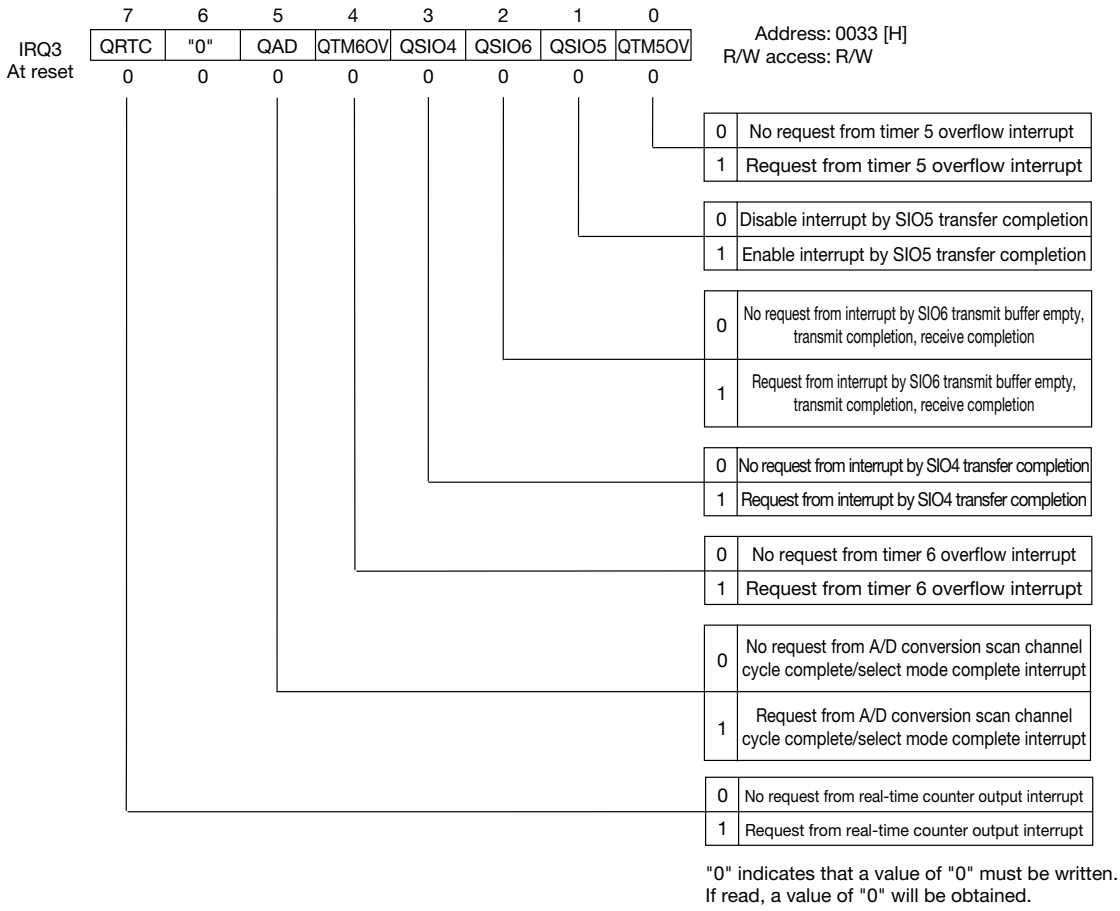


Figure 17-8 IRQ3 Configuration

(5) Interrupt request register 4 (IRQ4)

Interrupt request register 4 (IRQ4) consists of 5 bits. Bits are set to "1" corresponding to overflow of PWC0/matching of PWC0 and PWR0 (bit 0), overflow of PWC1/matching of PWC1 and PWR1 (bit 1), matching of PWC0 and PWR2 (bit 2), matching of PWC1 and PWR3 (bit 3), and overflow of timer 9 (bit 4).

IRQ4 can be read or written by the program. However, writes to bits 5 through 7 are invalid. If read, a value of "1" will always be obtained for bits 5 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ4 becomes E0H.

Figure 17-9 shows the configuration of IRQ4.

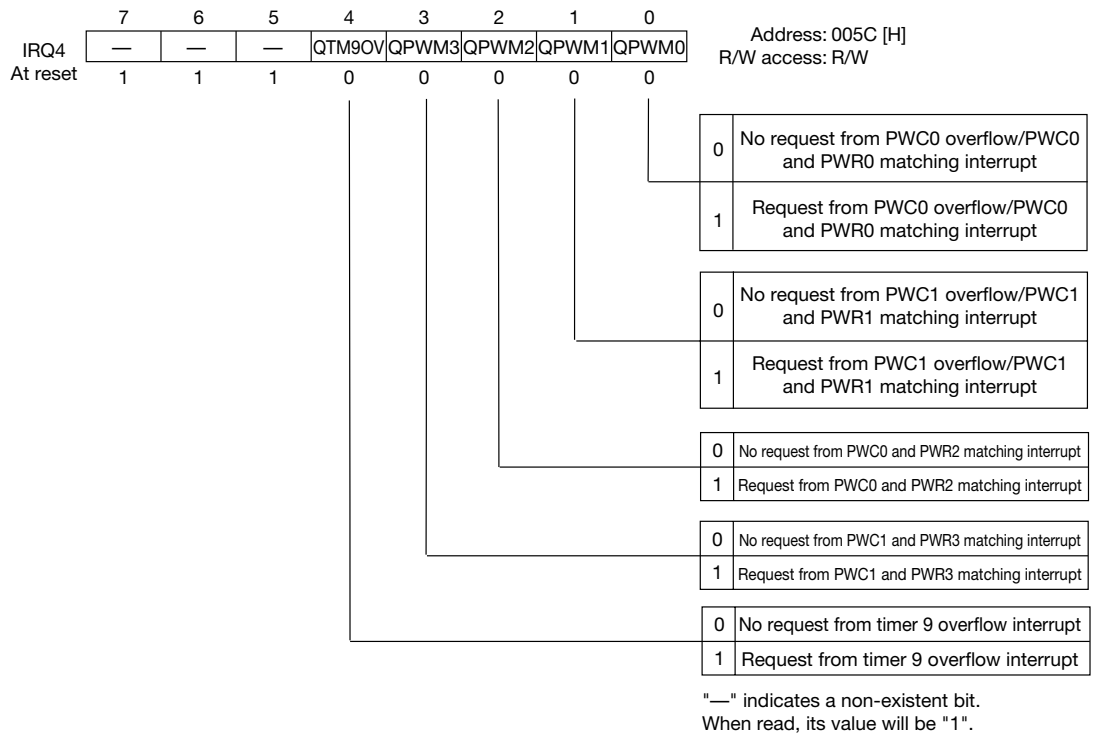


Figure 17-9 IRQ4 Configuration

17.4.2 Interrupt Enable Registers (IE0 to IE4)

(1) Interrupt enable register 0 (IE0)

Interrupt enable register 0 (IE0) consists of 4 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to external interrupt 0 (bit 0), overflow of free running counter (bit 1), CPCM0 event input /compare match (bit 6), and CPCM1 event input/ compare match (bit 7).

IE0 can be read or written by the program. However, if writing to bits 2 through 5, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 through 5.

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE0 becomes 00H.

Figure 17-10 shows the configuration of IE0.

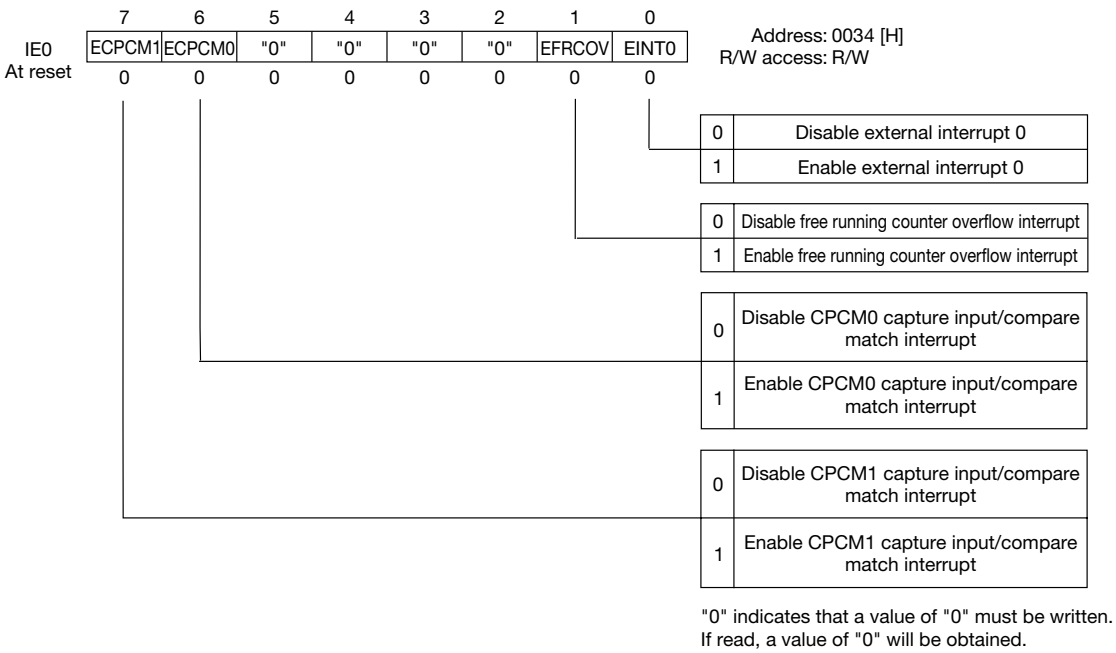


Figure 17-10 IE0 Configuration

(2) Interrupt enable register 1 (IE1)

Interrupt enable register 1 (IE1) consists of 7 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to overflow of timer 0 (bit 0), external interrupts 1 to 3 (bits 1 to 3), and overflow of timers 1 to 3 (bits 4 to 6).

IE1 can be read or written by the program. However, if writing to bit 7, always write the bit as "0". If read, a value of "0" will always be obtained for bit 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE1 becomes 00H.

Figure 17-11 shows the configuration of IE1.

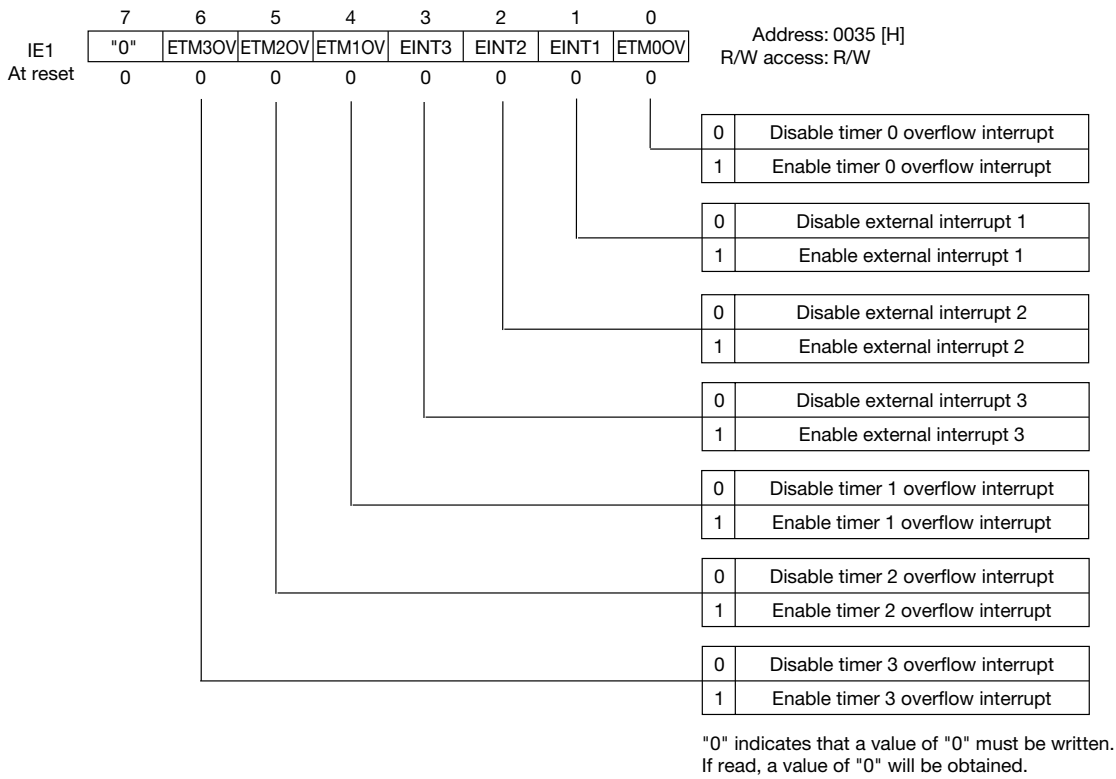


Figure 17-11 IE1 Configuration

(3) Interrupt enable register 2 (IE2)

Interrupt enable register 2 (IE2) consists of 6 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to external interrupts 4 to 7 (bits 0 to 3), overflow of timer 4 (bit 6), and SIO1 transmit buffer empty/transmit complete/receive complete (bit 7).

IE2 can be read or written by the program. However, if writing to bits 4 and 5, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4 and 5.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE2 becomes 00H.

Figure 17-12 shows the configuration of IE2.

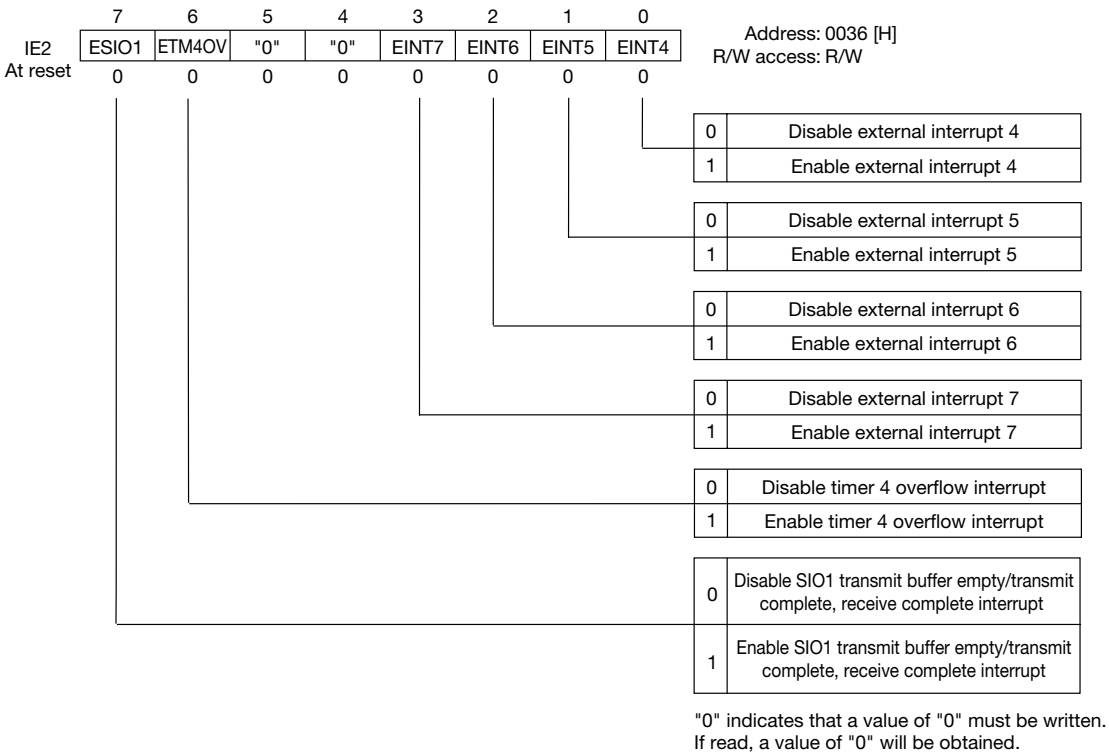


Figure 17-12 IE2 Configuration

(4) Interrupt enable register 3 (IE3)

Interrupt enable register 3 (IE3) consists of 7 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to overflow of timer 5 (bit 0), SIO5 and SIO4 transmit-receive completion (bits 1 and 3), SIO6 transmit buffer empty/transmit complete/receive complete (bit 2) overflow of timer 6 (bit 4), A/D conversion scan channel cycle complete/select mode complete (bit 5), and real-time counter output (bit 7).

IE3 can be read or written by the program. However, if writing to bit 6, always write the bit as "0". If read, a value of "0" will always be obtained for bit 6.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE3 becomes 00H.

Figure 17-13 shows the configuration of IE3.

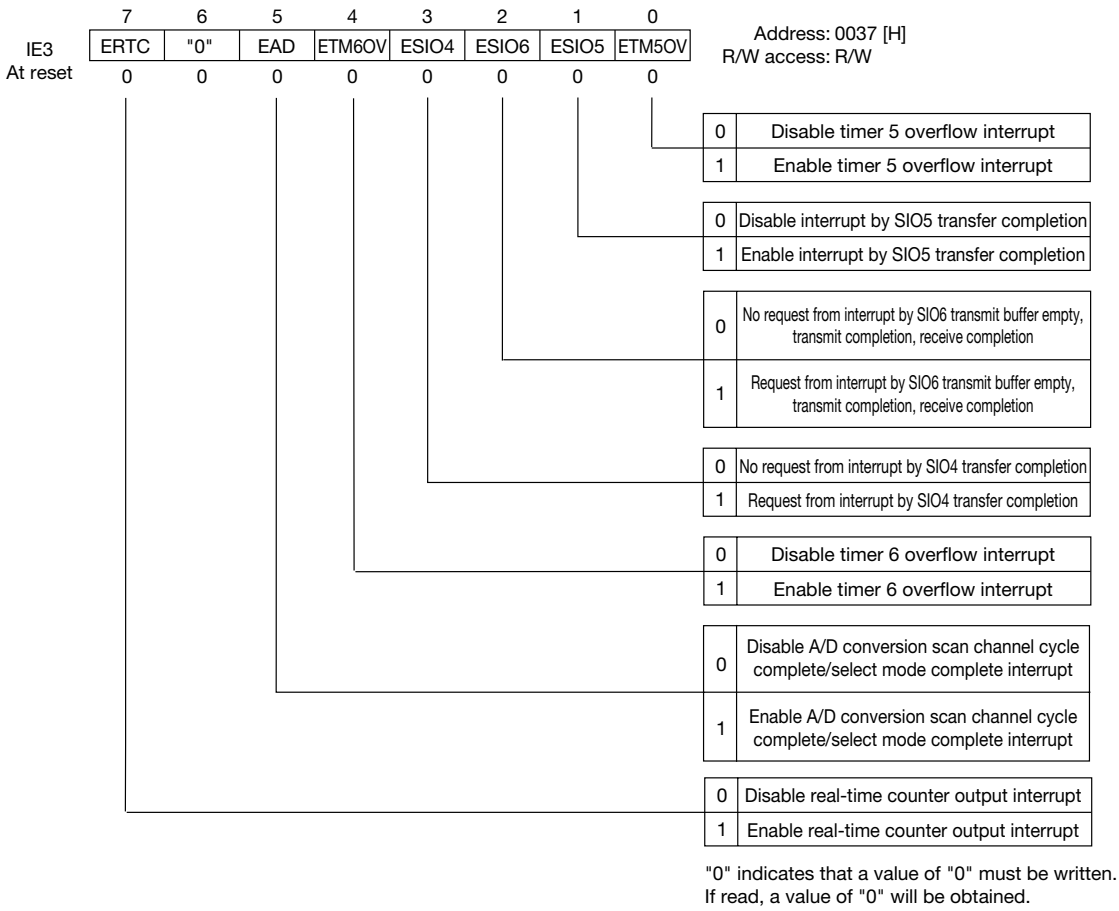


Figure 17-13 IE3 Configuration

(5) Interrupt enable register 4 (IE4)

Interrupt enable register 4 (IE4) consists of 5 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to overflow of PWC0/matching of PWC0 and PWR0 (bit 0), overflow of PWC1/matching of PWC1 and PWR1 (bit 1), matching of PWC0 and PWR2 (bit 2), matching of PWC1 and PWR3 (bit 3), and overflow of timer 9 (bit 4).

IE4 can be read or written by the program. However, writes to bits 5 through 7 are invalid. If read, a value of "1" will always be obtained for bits 5 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE4 becomes E0H.

Figure 17-14 shows the configuration of IE4.

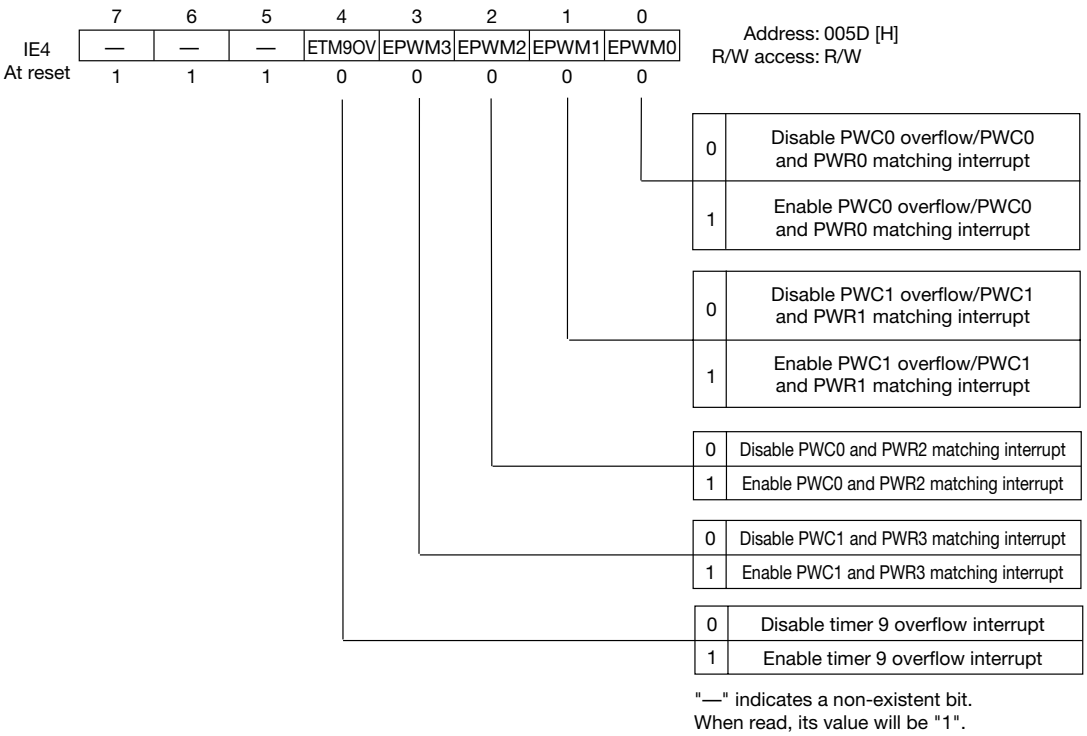


Figure 17-14 IE4 Configuration

17.4.3 Interrupt Priority Control Registers (IP0 to IP9)

(1) Interrupt priority control register 0 (IP0)

Interrupt priority control register 0 (IP0) consists of 4 bits and specifies the interrupt priority for external interrupt 0 (bits 0 and 1) and overflow of the free running counter (bits 2 and 3).

IP0 can be read or written by the program. However, if writing to bits 4 through 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP0 becomes 00H.

Figure 17-15 shows the configuration of IP0.

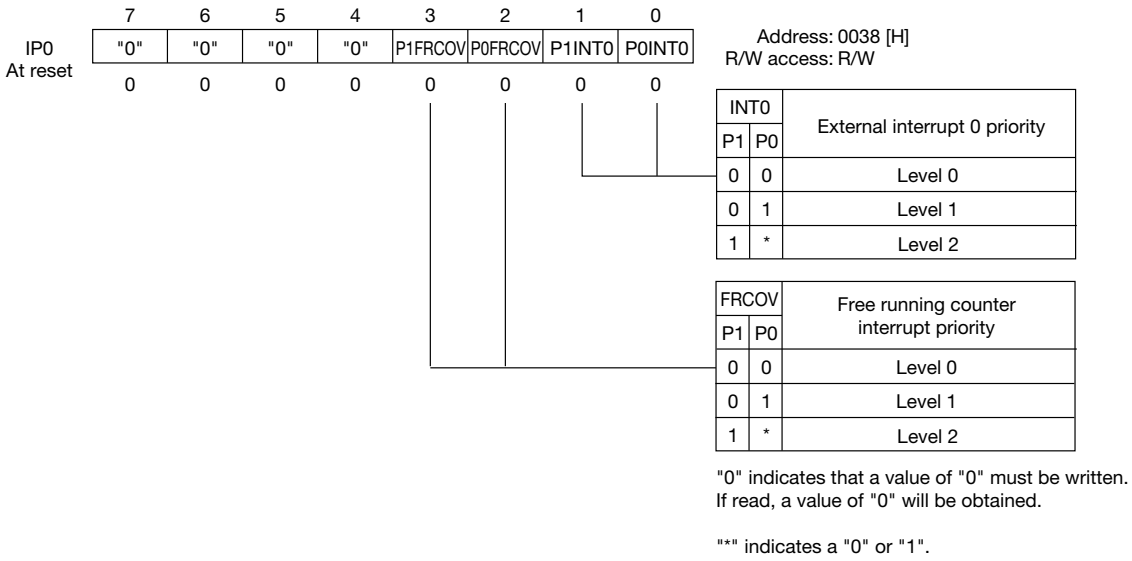


Figure 17-15 IP0 Configuration

(2) Interrupt priority control register 1 (IP1)

Interrupt priority control register 1 (IP1) consists of 4 bits and specifies interrupt priority for CPCM0 event input /compare match (bits 4 and 5) and CPCM1 event input/compare match (bits 6 and 7).

IP1 can be read or written by the program. However, if writing to bits 0 through 3, always write those bits as "0". If read, a value of "0" will always be obtained for bits 0 through 3.

When reset ($\overline{\text{RES}}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP1 becomes 00H.

Figure 17-16 shows the configuration of IP1.

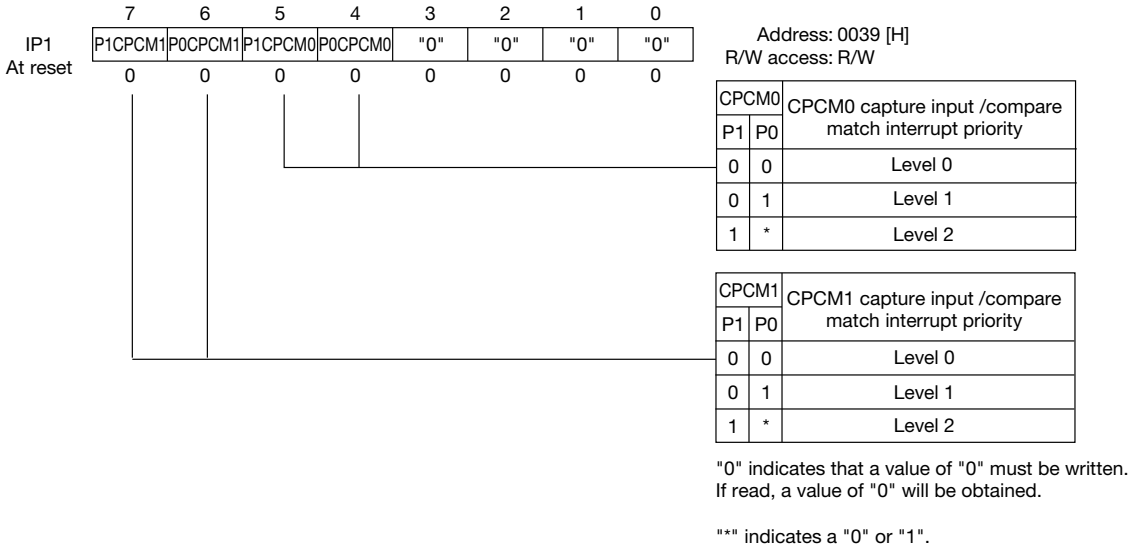


Figure 17-16 IP1 Configuration

(3) Interrupt priority control register 2 (IP2)

Interrupt priority control register 2 (IP2) consists of 8 bits and specifies interrupt priority for overflow of timer 0 (bits 0 and 1), external interrupts 1 (bits 2 and 3), external interrupt 2 (bits 4 and 5) and external interrupt 3 (bits 6 and 7).

IP2 can be read or written by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP2 becomes 00H.

Figure 17-17 shows the configuration of IP2.

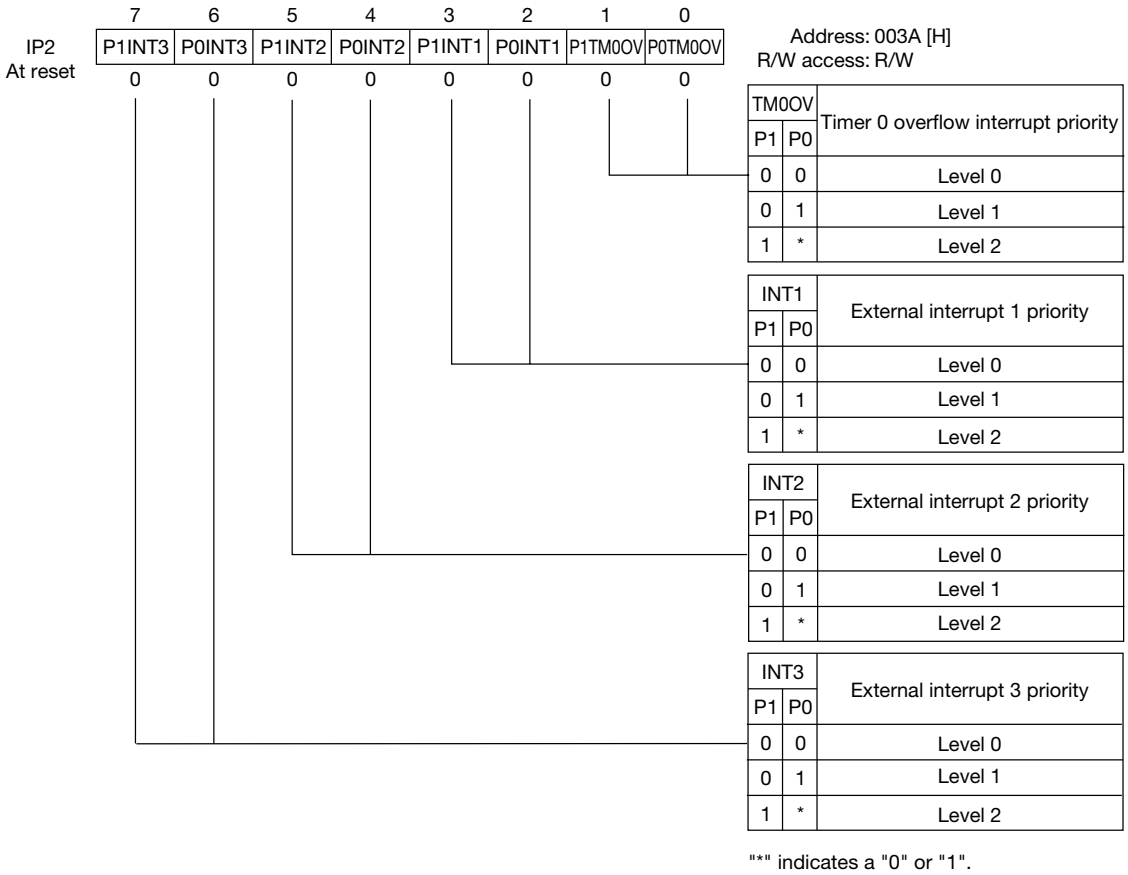


Figure 17-17 IP2 Configuration

(4) Interrupt priority control register 3 (IP3)

Interrupt priority control register 3 (IP3) consists of 6 bits and specifies interrupt priority for overflow of timers 1 to 3 (bits 0 to 5).

IP3 can be read or written by the program. However, if writing to bits 6 and 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 6 and 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP3 becomes 00H.

Figure 17-18 shows the configuration of IP3.

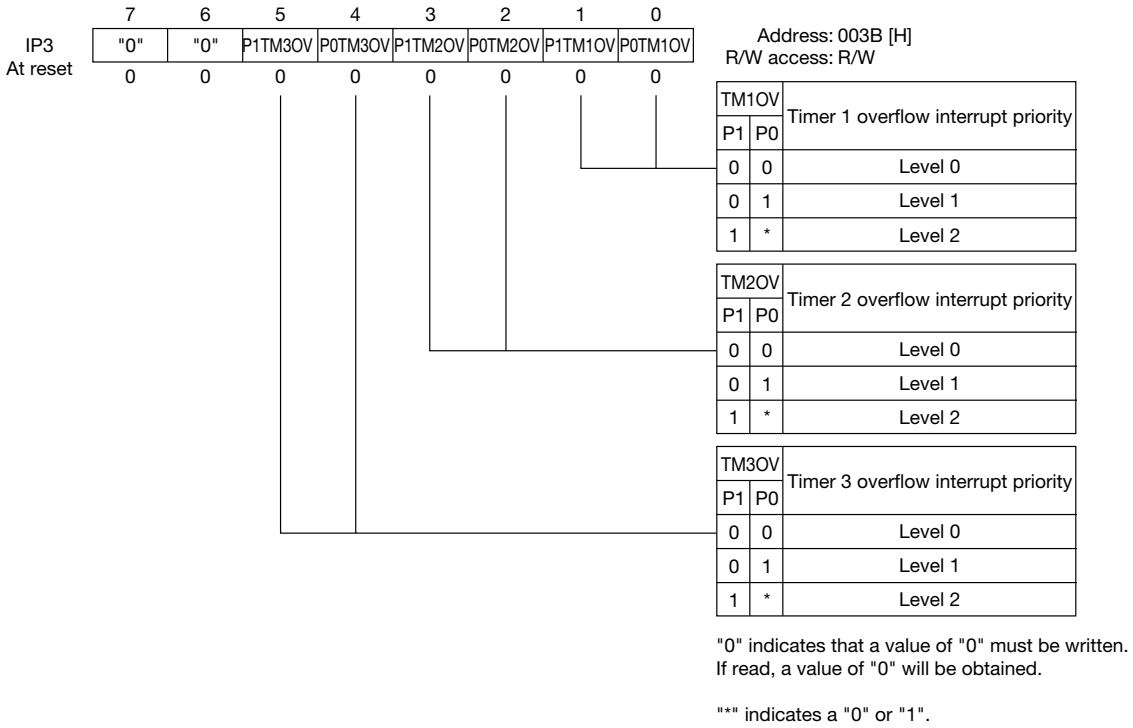


Figure 17-18 IP3 Configuration

(5) Interrupt priority control register 4 (IP4)

Interrupt priority control register 4 (IP4) consists of 8 bits and specifies interrupt priority for external interrupt 4 (bits 0 and 1), external interrupt 5 (bits 2 and 3), external interrupt 6 (bits 4 and 5) and external interrupt 7 (bits 6 and 7).

IP4 can be read or written by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP4 becomes 00H.

Figure 17-19 shows the configuration of IP4.

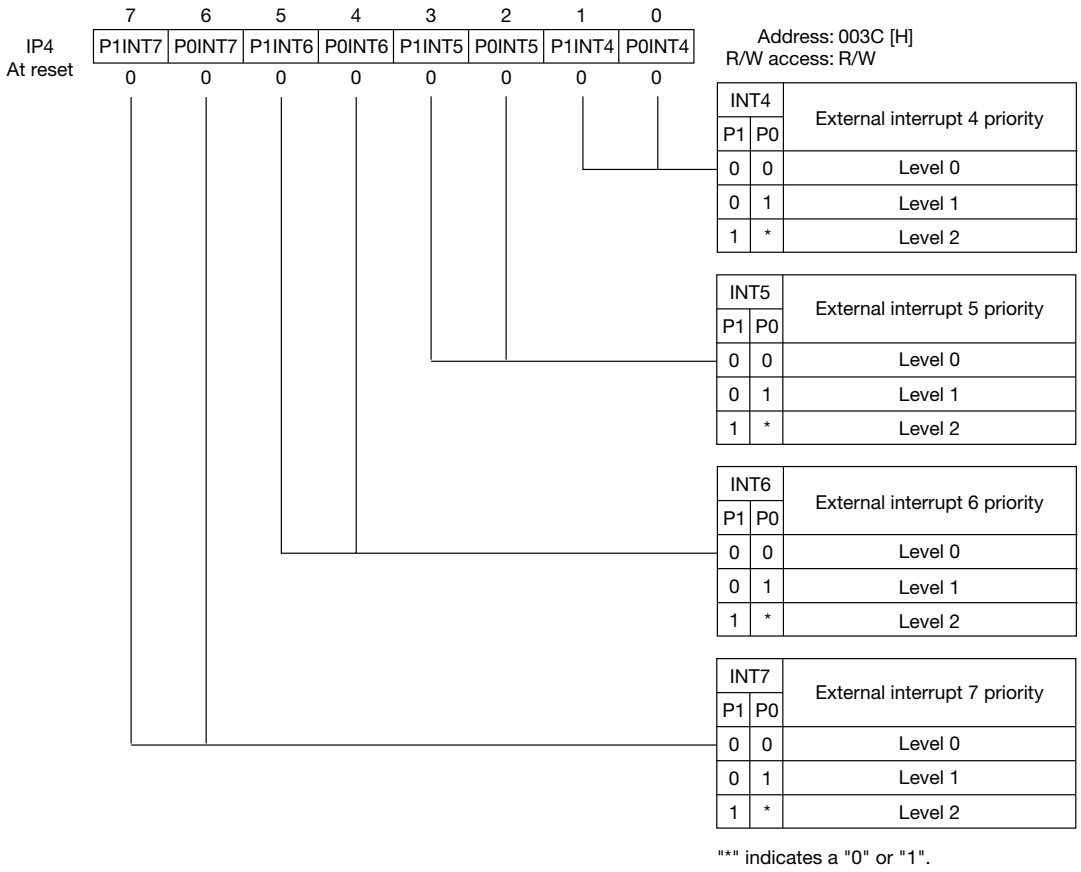


Figure 17-19 IP4 Configuration

(6) Interrupt priority control register 5 (IP5)

Interrupt priority control register 5 (IP5) consists of 4 bits and specifies interrupt priority for overflow of timer 4 (bits 4 and 5) and SIO1 transmit buffer empty/transmit complete/receive complete (bits 6 and 7).

IP5 can be read or written by the program. However, if writing to bits 0 through 3, always write those bits as "0". If read, a value of "0" will always be obtained for bits 0 through 3.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP5 becomes 00H.

Figure 17-20 shows the configuration of IP5.

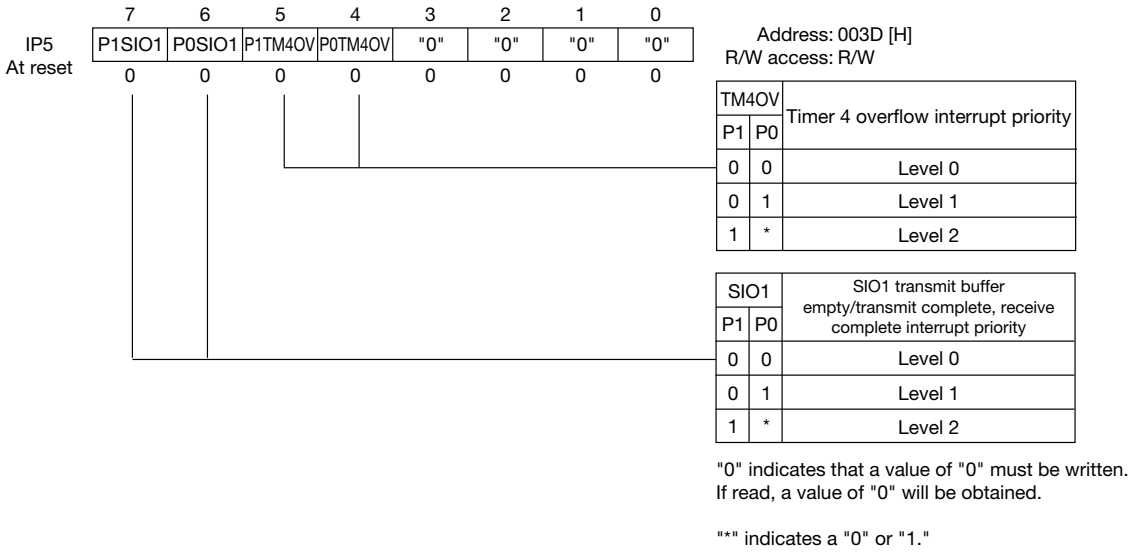


Figure 17-20 IP5 Configuration

(7) Interrupt priority control register 6 (IP6)

Interrupt priority control register 6 (IP6) consists of 8 bits and specifies interrupt priority for overflow of timer 5 (bits 0 and 1), SIO6 transmit buffer empty/transmit coplete/receive complete (bits 4 and 5) and SIO5 and SIO4 transmit-receive completion (bits 2, 3, 6 and 7).

IP6 can be read or written by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP6 becomes 00H.

Figure 17-21 shows the configuration of IP6.

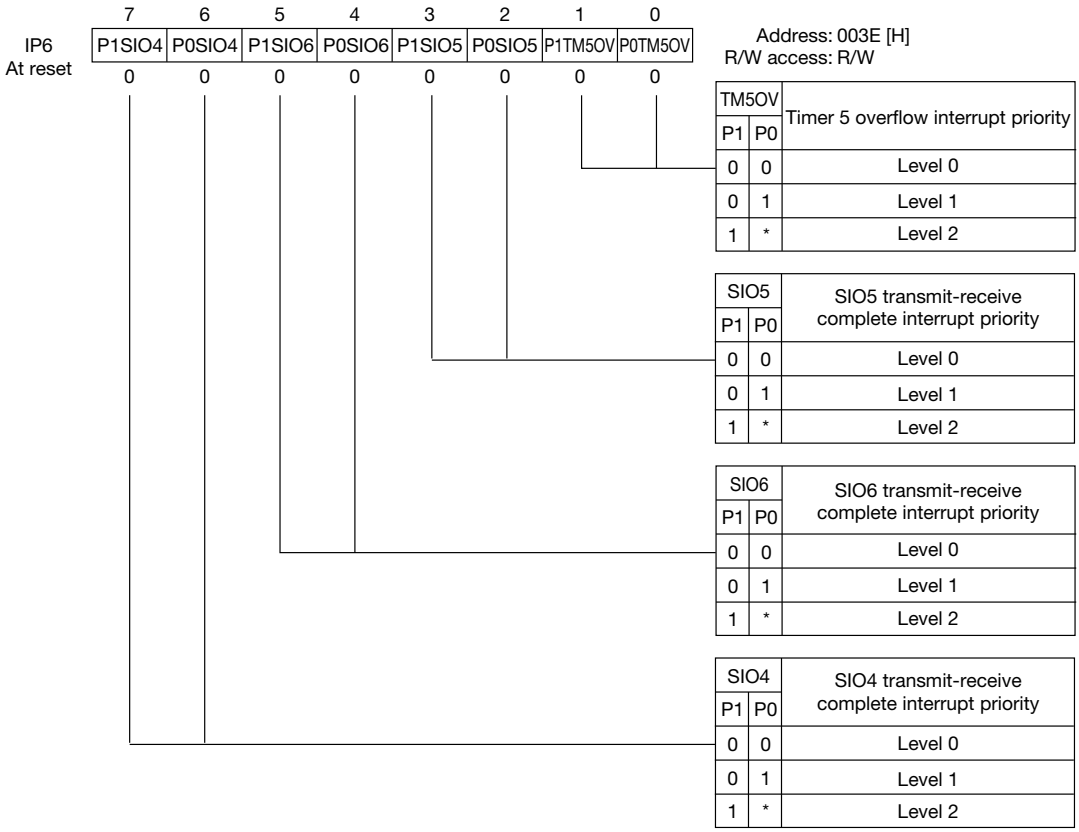


Figure 17-21 IP6 Configuration

(8) Interrupt priority control register 7 (IP7)

Interrupt priority control register 7 (IP7) consists of 6 bits and specifies interrupt priority for overflow of timer 6 (bits 0 and 1), A/D conversion scan channel cycle complete/select mode complete (bits 2 and 3), and real-time counter output (bits 6 and 7).

IP7 can be read or written by the program. However, if writing to bits 4 and 5, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4 and 5.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP7 becomes 00H.

Figure 17-22 shows the configuration of IP7.

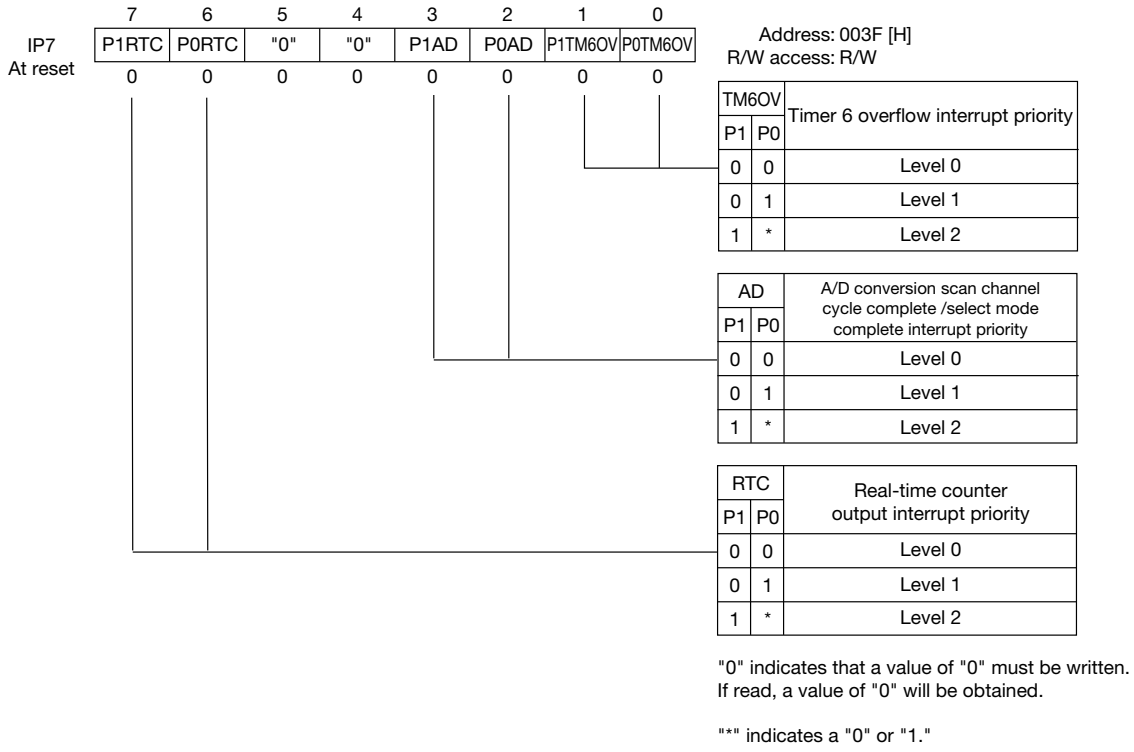


Figure 17-22 IP7 Configuration

(9) Interrupt priority control register 8 (IP8)

Interrupt priority control register 8 (IP8) consists of 8 bits and specifies interrupt priority for overflow of PWC0/matching of PWC0 and PWR0 (bits 0 and 1), overflow of PWC1/ matching of PWC1 and PWR1 (bits 2 and 3), matching of PWC0 and PWR2 (bits 4 and 5) and matching of PWC1 and PWR3 (bits 6 and 7).

IP8 can be read or written by the program.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP8 becomes 00H.

Figure 17-23 shows the configuration of IP8.

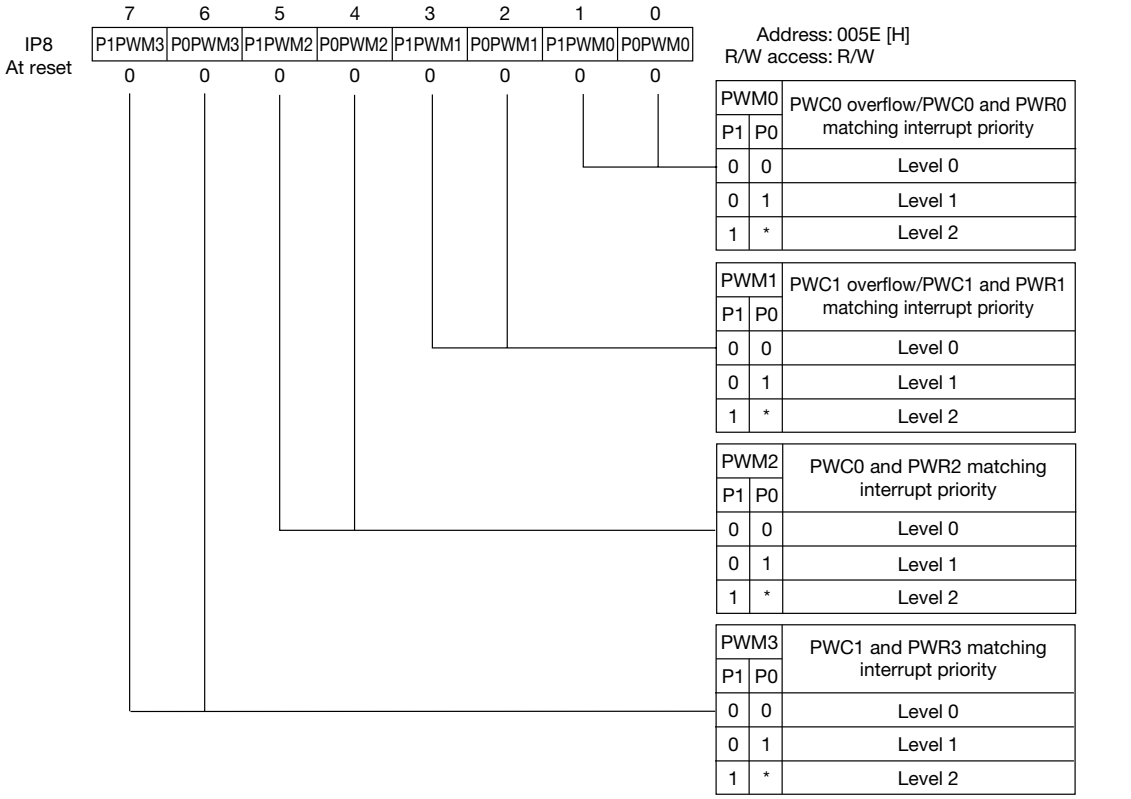


Figure 17-23 IP8 Configuration

(10) Interrupt priority control register 9 (IP9)

Interrupt priority control register 9 (IP9) consists of 2 bits and specifies interrupt priority for the overflow of timer 9 (bits 0 and 1).

IP9 can be read or written by the program. However, writes to bits 2 through 7 are invalid. If read, a value of "1" will always be obtained for bits 2 through 7.

When reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP9 becomes FCH.

Figure 17-24 shows the configuration of IP9.

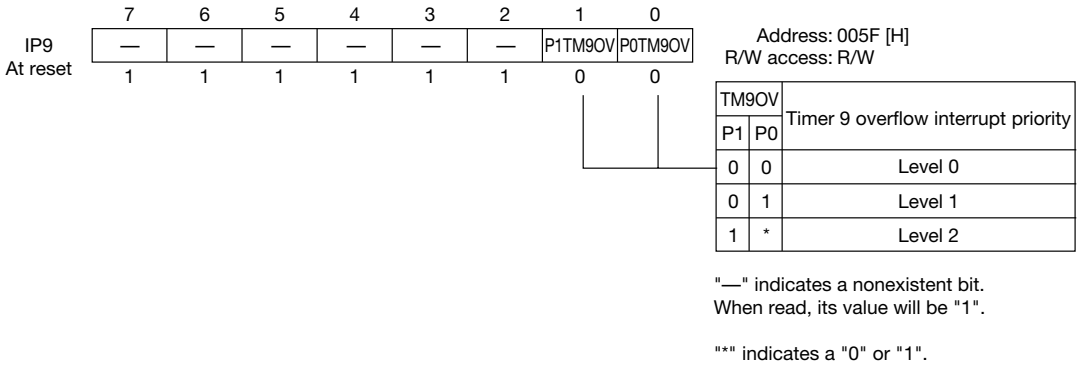


Figure 17-24 IP9 Configuration

Chapter 18

Bus Port Functions

18 Bus Port Functions

18.1 Overview

The MSM66577 family can externally expand program memory (usually ROM) up to a maximum of 1MB and data memory (usually RAM) up to a maximum of 1MB. With the SELMBUS pin setting, a multiplexed bus type (address/data time division) or a separate bus type (independent address/data buses) can be selected for the bus port interface of the MSM66577 family.

Bus ports (A0 to A19, D0 to D7) and control signals (SELMBUS, ALE, $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$) are used to access the external program memory and external data memory.

When the SELMBUS pin is at a high level, bus ports are configured as the separate bus with 20 address (A0 to A19) lines and 8 data (D0 and D7) lines and assigned as the secondary functions of port 0 (P0), port 1 (P1), port 2 (P2) and port 4 (P4). Unnecessary upper addresses can be reset as normal I/O ports.

$\overline{PSEN}$ (P3_1) is used as a strobe signal to read the external program memory. $\overline{RD}$ (P3_2) and $\overline{WR}$ (P3_3) are used as read and write strobes for external data memory.

When the SELMBUS pin is at a low level, bus ports are configured as the multiplexed bus with 12 address (A8 to A19) lines and 8 address/data combined lines (D0 to D7) and assigned as the secondary functions of port 0 (P0), port 1 (P1), and port 2 (P2). Unnecessary upper addresses can be reset as normal I/O ports.

18.2 Port Operation

18.2.1 Port Operation When Accessing Program Memory

- **Separate bus type (the SELMBUS pin at a high level)**

When accessing internal program memory (addresses 0H to 1FFFFH with the $\overline{EA}$ pin at a high level), P0, P1, P2, P3_1 and P4 operate as I/O ports.

When accessing external program memory (the $\overline{EA}$ pin at a low level or addresses 20000H to FFFFFH with the $\overline{EA}$ pin at a high level), P0 operates as the program data input port, P1, P2, and P4 operate as address output ports, and P3_1 operates as the $\overline{PSEN}$ output port.

If the $\overline{EA}$ pin is at a low level, P0, P1, P2, P3_1 and P4 are automatically switched (secondary function control registers and mode registers are set) to bus port and control signal functions (hereafter referred to as bus port functions) when reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap). If the $\overline{EA}$ pin is at a high level, bus port functions are not automatically switched. It is necessary to switch to bus port functions before external program memory is accessed by setting secondary function control registers and mode registers on software.

Of the ports that are automatically set as bus port functions when the $\overline{EA}$ pin is at a low level, if upper address or other output is unnecessary, then after reset, those ports can be operated as I/O ports by resetting their secondary function control register.

Table 18-1 lists the operation of P0, P1, P2, P3_1 and P4 in the separate bus type during a program memory access.

Table 18-1 P0, P1, P2, P3_1 and P4 Operation During Program Memory Access (Separate Bus Type)

Memory to be accessed	Address	P0 operation	P1, P2, P4 operation	P3_1 operation
Internal program	When $\overline{EA}$ = H, 0H to 1FFFFH	I/O port		
External program	When $\overline{EA}$ = H, 20000H to FFFFFH	After set as secondary function output, program data input	After set as secondary function output, address output	After set as secondary function output, $\overline{PSEN}$ output
	When $\overline{EA}$ = L, 0H to FFFFFH	Program data input	Address output	$\overline{PSEN}$ output

- **Multiplexed bus type (the SELMBUS pin at a low level)**

  When accessing internal program memory (addresses 0H to 1FFFFH with the $\overline{EA}$ pin at a high level), P0, P1, P2, P3_1 and P4 operate as I/O ports.

  When accessing external program memory (the $\overline{EA}$ pin at a low level or addresses 20000H to FFFFFH with the $\overline{EA}$ pin at a high level), P0 operates as the address output and program data input port, P1 and P2 operate as addresses output ports, P3_0 operates as the ALE output port, and P3_1 operates as the $\overline{PSEN}$ output port.

  If the $\overline{EA}$ pin is at a low level, P0, P1, P2, P3_0 and P3_1 are automatically switched (secondary function control registers and mode registers are set) to bus port and control signal functions (hereafter referred to as bus port functions) when reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap). If the $\overline{EA}$ pin is at a high level, bus port functions are not automatically switched. It is necessary to switch to bus port functions before external program memory is accessed by setting secondary function control registers and mode registers on software.

  Of the ports that are automatically set as bus port functions when the $\overline{EA}$ pin is at a low level, if upper address or other output is unnecessary, then after reset, those ports can be operated as normal I/O ports by resetting their secondary function control register.

  Table 18-2 lists the operation of P0, P1, P2, P3_0 and P3_1 in the multiplexed bus type during a program memory access.

Table 18-2 P0, P1, P2, P3_0 and P3_1 Operation During Program Memory Access (Multiplexed Bus Type)

Memory to be accessed	Address	P0 operation	P1, P2, P4 operation	P3_1 operation	P3_0 operation
Internal program	When $\overline{EA}$ = H, 0H to 1FFFFH	I/O port			
External program	When $\overline{EA}$ = H, 20000H to FFFFFH	After set as secondary function output, address output/program data input	After set as secondary function, address output	After set as secondary function output, $\overline{PSEN}$ output	After set as secondary function output, ALE output
	When $\overline{EA}$ = L, 0H to FFFFFH	Address output/program data input	Address output	$\overline{PSEN}$ output	ALE output

18.2.2 Port Operation When Accessing Data Memory

- **Separate bus type (the SELMBUS pin at a high level)**

When accessing internal data memory (addresses 0H to 11FFH), P0, P1, P2, P3_2, P3_3 and P4 operate as I/O ports.

When accessing external data memory (addresses 1200H to FFFFFH), set ports P0, P1, P2, P3_2, P3_3 and P4 to their secondary functions so that P0 operates as a data I/O pin, P1, P2, and P4 operate as address output pins, and P3_2 and P3_3 operate as $\overline{RD}$ and $\overline{WR}$ output pins.

If the $\overline{EA}$ pin is at a low level, P0, P1, P2 and P4 are automatically set as bus ports (secondary function control registers and mode registers are set) when reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap). Because P3_2 and P3_3 are automatically set as input ports instead of $\overline{RD}$ and $\overline{WR}$ output pins, before external data memory is accessed, they must be set as secondary function outputs.

Of the ports that are automatically set as bus port functions when the $\overline{EA}$ pin is at a low level, if upper address or other output is unnecessary, then after reset, those ports can be operated as I/O ports by resetting their secondary function control register.

Table 18-3 lists the operation of P0, P1, P2, P3_2, P3_3 and P4 in the separate bus type during a data memory access.

Table 18-3 P0, P1, P2, P3_2, P3_3 and P4 Operation During Data Memory Access (Separate Bus Type)

Data to be accessed	Address	P0 operation	P1, P2, P4 operation	P3_2, P3_3 operation
Internal data	0H to 11FFH	I/O port		
External data	1200H to FFFFFH	After set as secondary function output *1, data I/O	After set as secondary function output *1, address output	After set as secondary function output, $\overline{RD}$ and $\overline{WR}$ output

*1 If the $\overline{EA}$ pin is at a low level, P0, P1, P2 and P4 are automatically set as secondary function outputs when reset.

- **Multipexed bus type (the SELMBUS pin at a low level)**

  When accessing internal data memory (addresses 0H to 11FFH), P0, P1, P2, P3_0, P3_2 and P3_3 operate as I/O ports.

  When accessing external data memory (addresses 1200H to FFFFFH), set ports P0, P1, P2, P3_0, P3_2 and P3_3 to their secondary functions so that P0 operates as an address output and data I/O pin, P1 and P2 operate as address output pins, P3_0, P3_2 and P3_3 operate as ALE, $\overline{RD}$ and $\overline{WR}$ output pins.

  If the $\overline{EA}$ pin is at a low level, P0, P1 and P2 are automatically set as bus ports (secondary function control registers and mode registers are set) when reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap). Because P3_2 and P3_3 are automatically set as input ports instead of $\overline{RD}$ and $\overline{WR}$ output pins, before external data memory is accessed, they must be set as secondary function outputs.

  Of the ports that are automatically set as bus port functions when the $\overline{EA}$ pin is at a low level, if upper address or other output is unnecessary, then after reset, those ports can be operated as I/O ports by resetting their secondary function control register.

  Table 18-4 lists the operation of P0, P1, P2, P3_0, P3_2 and P3_3 in the multiplexed bus type during a data memory access.

Table 18-4 P0, P1, P2, P3_0, P3_2 and P3_3 Operation During Data Memory Access (Multiplexed Bus Type)

Data to be accessed	Address	P0 operation	P1, P2 operation	P3_0, P3_2, P3_3 operation
Internal data	0H to 11FFH	I/O port		
External data	1200H to FFFFFH	After set as secondary function output *1, data I/O	After set as secondary function output *1, address output	After set as secondary function output, ALE, $\overline{RD}$ and $\overline{WR}$ output

*1 If the $\overline{EA}$ pin is at a low level, P0, P1 and P2 are automatically set as secondary function outputs when reset.

18.3 External Memory Access

18.3.1 External Program Memory Access

- **Separate bus type (the SELMBUS pin at a high level)**

  A program memory space of 1MB maximum (00000H to FFFFFH) can be accessed with the 16-bit program counter (PC) and 4-bit code segment register (CSR). When the $\overline{EA}$ pin is set to a high level, program addresses from 10000H to FFFFFH access external program memory. When the $\overline{EA}$ pin is set to a low level, program addresses from 00000H to FFFFFH access external program memory.

  If the $\overline{EA}$ pin is set to a high level and external program memory is to be used from 10000H to FFFFFH, then P0, P1, P2 and P4 must be set as secondary function outputs. In addition, P3_1 ($\overline{PSEN}$ output) must also be set as a secondary function output.

  Figure 18-1 shows an example connection of external program memory (ROM) in the separate bus type.

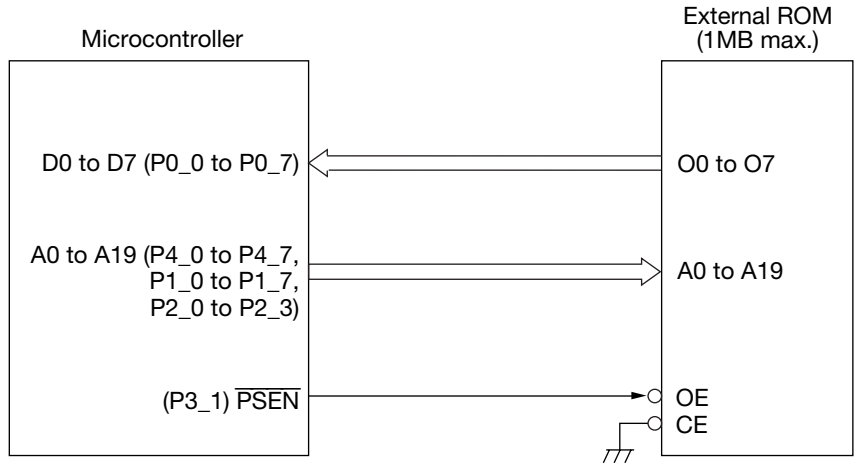


Figure 18-1 External ROM Connection Example (Separate Bus Type)

- **Multiplexed bus type (the SELMBUS pin at a low level)**

  A program memory space of 1MB maximum (00000H to FFFFFH) can be accessed with the 16-bit program counter (PC) and 4-bit code segment register (CSR). When the $\overline{EA}$ pin is set to a high level, program addresses from 10000H to FFFFFH access external program memory. When the $\overline{EA}$ pin is set to a low level, program addresses from 00000H to FFFFFH access external program memory.

  If the $\overline{EA}$ pin is set to a high level and external program memory is to be used from 10000H to FFFFFH, then P0, P1 and P2 must be set as secondary function outputs. In addition, P3_0 and P3_1 ($\overline{PSEN}$ output) must also be set as a secondary function output.

  Figure 18-2 shows an example connection of external program memory (ROM) in the multiplexed bus type.

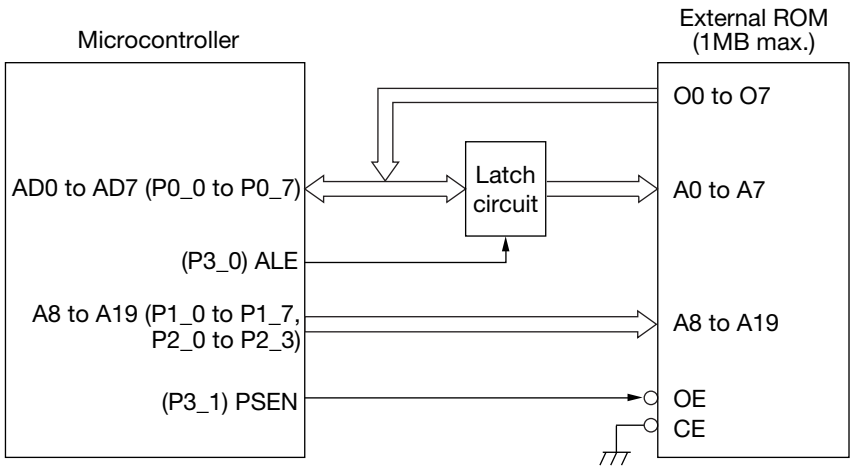


Figure 18-2 External ROM Connection Example (Multiplexed Bus Type)

18.3.2 External Data Memory Access

- **Separate bus type (the SELMBUS pin at a high level)**

A data memory space of 1MB maximum (0000H to FFFFFH) can be accessed with the 16-bit RAM address pointer (RAP) and 4-bit data segment register (DSR). Data addresses from 0000H to 11FFH access internal data memory. Data addresses from 1200H to FFFFFH access external data memory. External data memory is accessed in 8-bit (byte) units.

If external data memory is to be used, P0 must be set as a secondary function output (memory data I/O). Also, corresponding to the memory address, P1, P2 and P4 must be set as secondary function outputs (address outputs). In addition, P3_2 and P3_3 must be set as secondary function outputs ($\overline{WR}$ and $\overline{RD}$ outputs).

If the $\overline{EA}$ pin is at a low level, P0, P1, P2 and P4 automatically become secondary function outputs.

If necessary, insert external pull-up resistors at the $\overline{WR}$ and $\overline{RD}$ pins.

Figure 18-3 shows an example connection of external data memory (RAM) in the separate bus type.

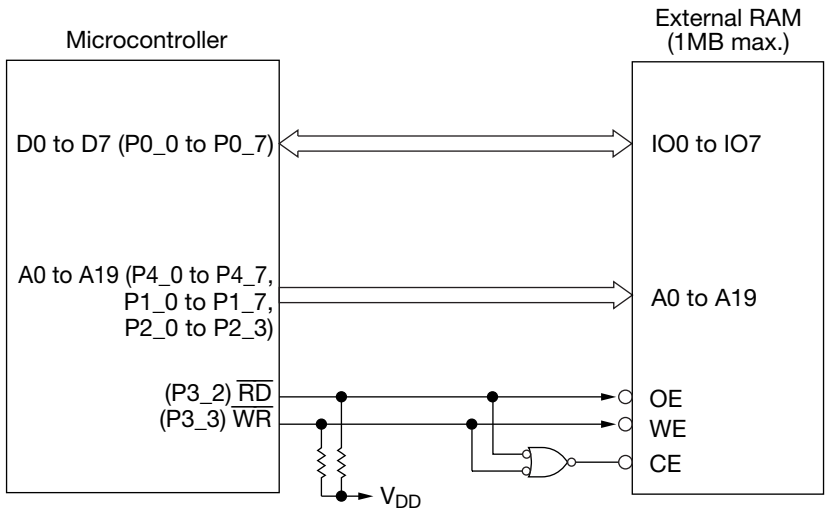


Figure 18-3 External RAM Connection Example (Separate Bus Type)

- **Multiplexed bus type (the SELMBUS pin at a low level)**

  A data memory space of 1MB maximum (0000H to FFFFFH) can be accessed with the 16-bit RAM address pointer (RAP) and 4-bit data segment register (DSR). Data addresses from 0000H to 11FFH access internal data memory. Data addresses from 1200H to FFFFFH access external data memory. External data memory is accessed in 8-bit (byte) units.

  If external data memory is to be used, P0 must be set as a secondary function output (memory data I/O). Also, corresponding to the memory address, P1 and P2 must be set as secondary function outputs (address outputs). In addition, P3_0, P3_2 and P3_3 must be set as secondary function outputs ($\overline{WR}$ and $\overline{RD}$ outputs).

  If the $\overline{EA}$ pin is at a low level, P0, P1, P2 and P4 automatically become secondary function outputs.

  If necessary, insert external pull-up resistors at the $\overline{WR}$ and $\overline{RD}$ pins.

  Figure 18-4 shows an example connection of external data memory (RAM) in the multiplexed bus type.

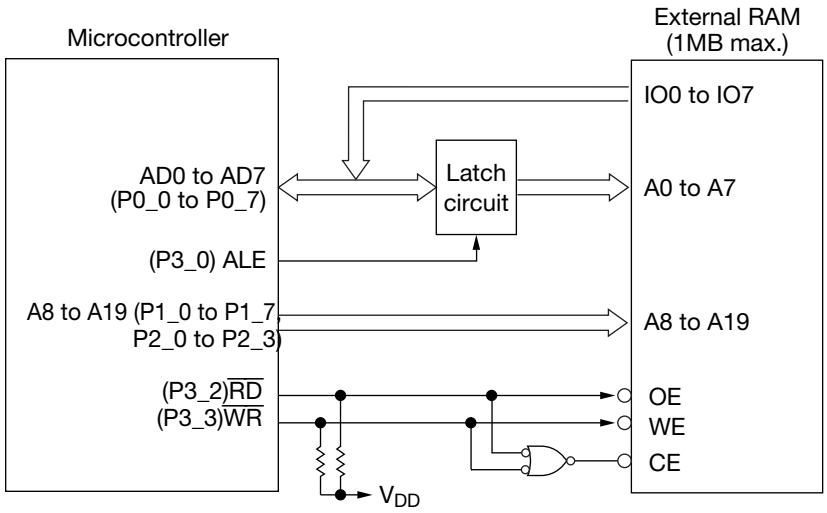


Figure 18-4 External RAM Connection Example (Multiplexed Bus Type)

18.4 External Memory Access Timing

18.4.1 External Program Memory Access Timing

Figures 18-5 and 18-6 show the timing for accessing external program memory in the separate bus type.

Figure 18-7 and 18-8 show the timing for accessing external program memory in the multiplexed bus type.

For external memory with slow access times, a function is available to insert wait cycles (see Section 4.4, "READY Function"). Use this function to match the access time of the external memory to be used. The ROMRDY register specifies the number of wait cycles to insert.

For actual AC characteristics, refer to Chapter 20, "Electrical Characteristics".

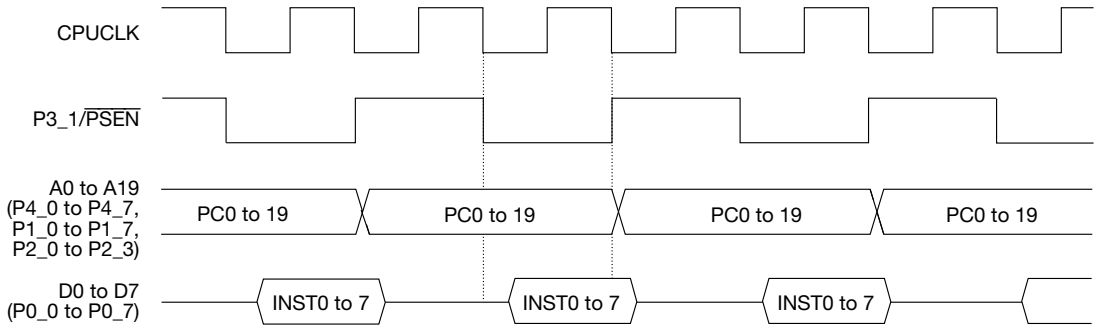


Figure 18-5 External Program Memory Access Timing (No Wait Cycles) (Separate Bus Type)

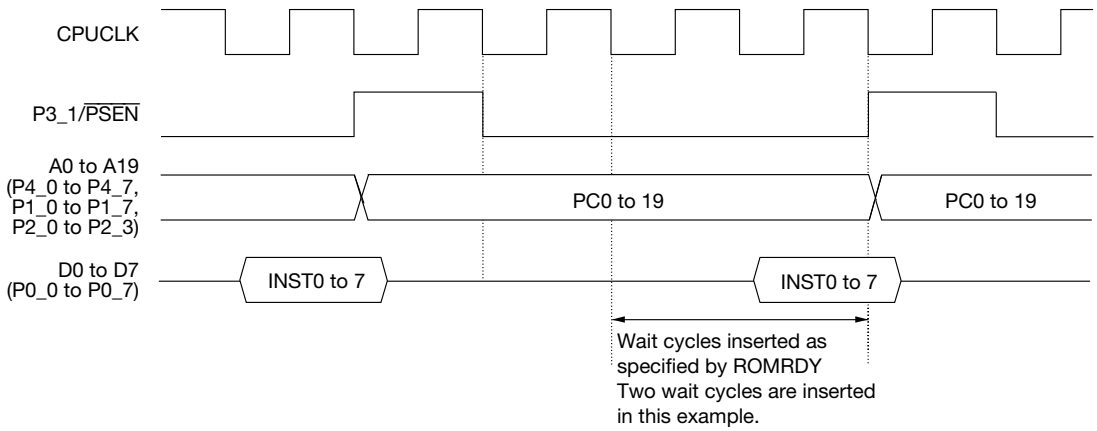


Figure 18-6 External Program Memory Access Timing (2 Wait Cycles) (Separate Bus Type)

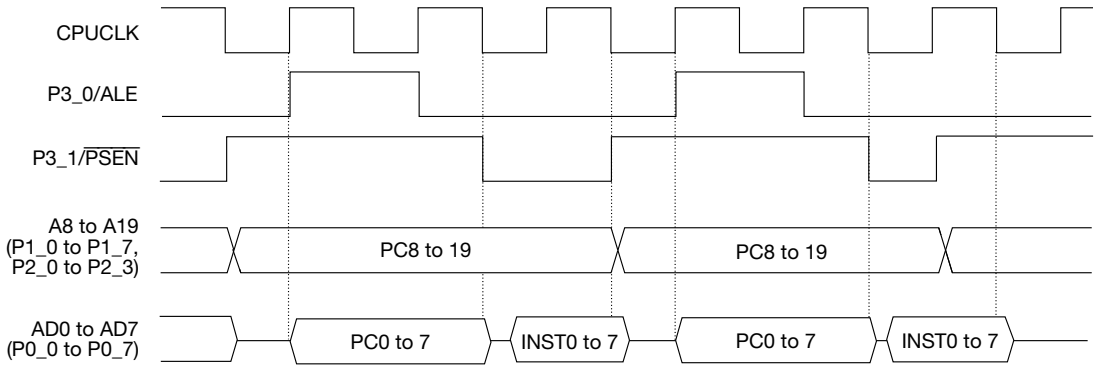


Figure 18-7 External Program Memory Access Timing (No Wait Cycles) (Multiplexed Bus Type)

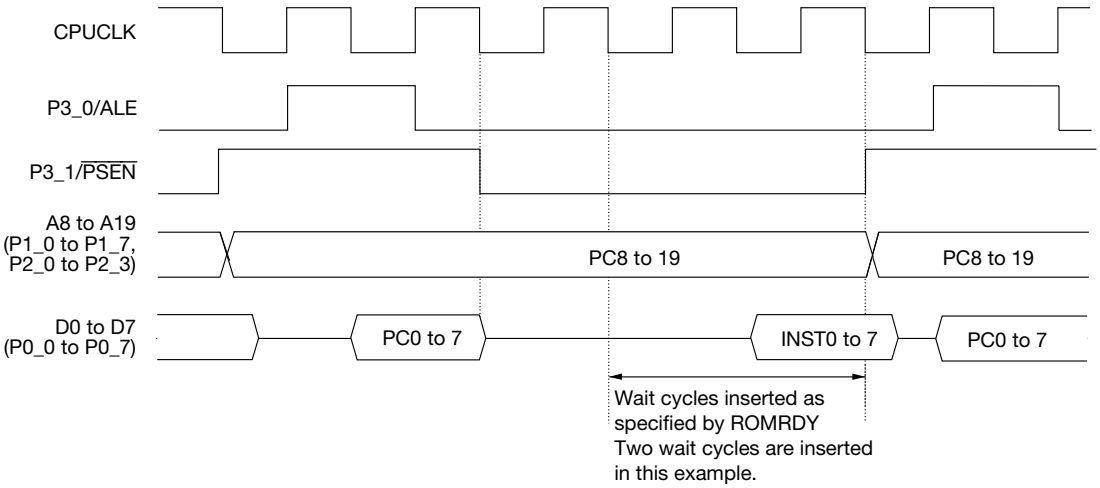


Figure 18-8 External Program Memory Access Timing (2 Wait Cycles) (Multiplexed Bus Type)

18.4.2 External Data Memory Access Timing

Figures 18-9 and 18-10 show the timing for accessing data program memory in the separate bus type.

For external memory with slow access times, a function is available to insert wait cycles (see section 4.4, "Ready Function"). Use this function to match the access time of the external memory to be used. Compared to internal data memory accesses, when accessing external data memory, 2 or 3 wait cycles are automatically inserted for each 1 byte access. The RAMRDY register specifies the number of wait cycles to insert in addition to the 3 to 4 cycles that are automatically inserted.

For actual AC characteristics, refer to Chapter 20, "Electrical Characteristics".

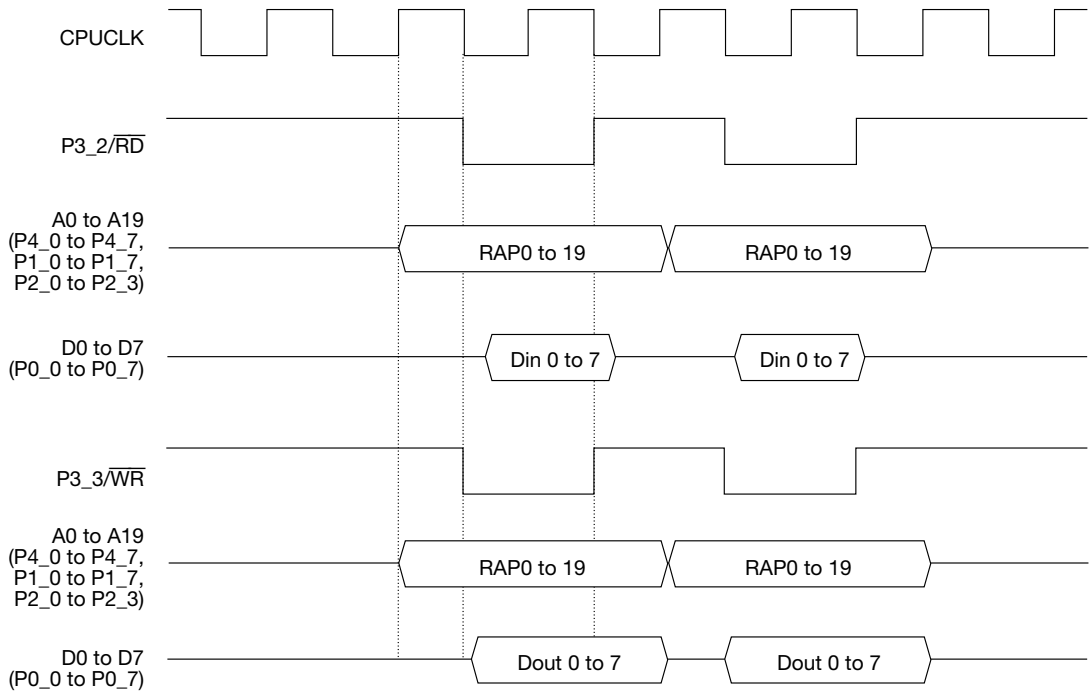


Figure 18-9 External Data Memory Access Timing (Word Access: No Wait Cycles) (Separate Bus Type)

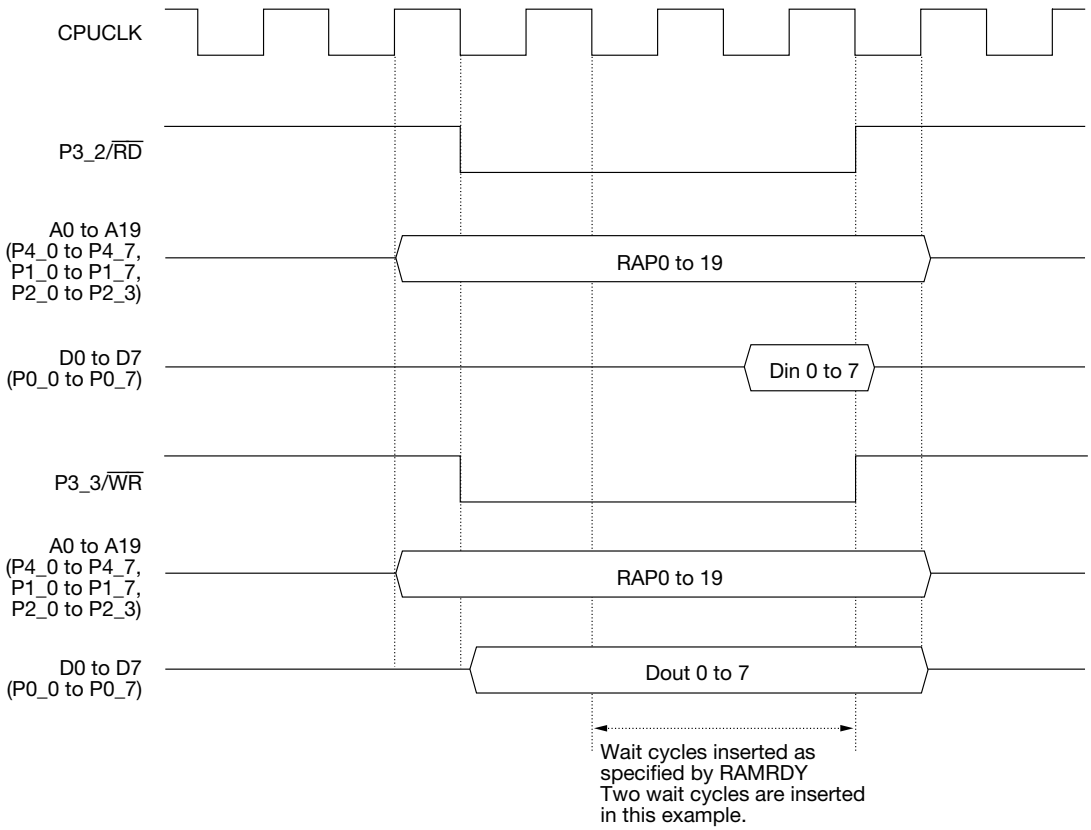


Figure 18-10 External Data Memory Access Timing (Byte Access: 2 Wait Cycles) (Separate Bus Type)

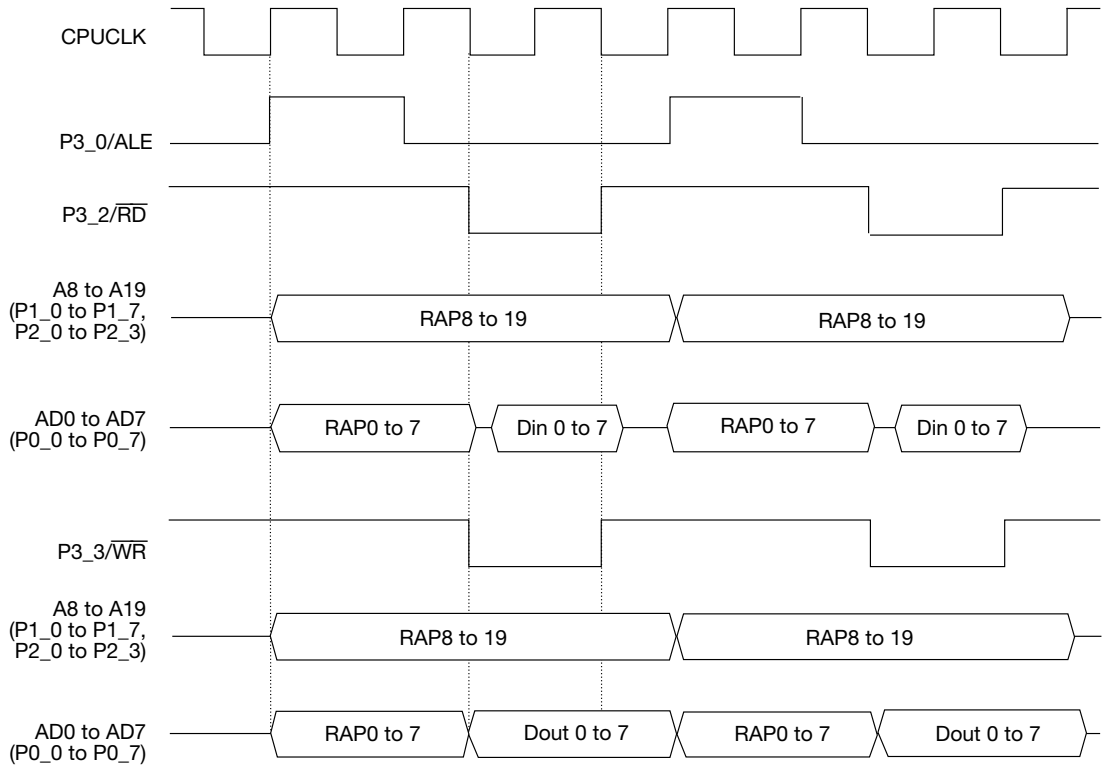


Figure 18-11 External Data Memory Access Timing (Word Access: No Wait Cycles) (Multiplexed Bus Type)

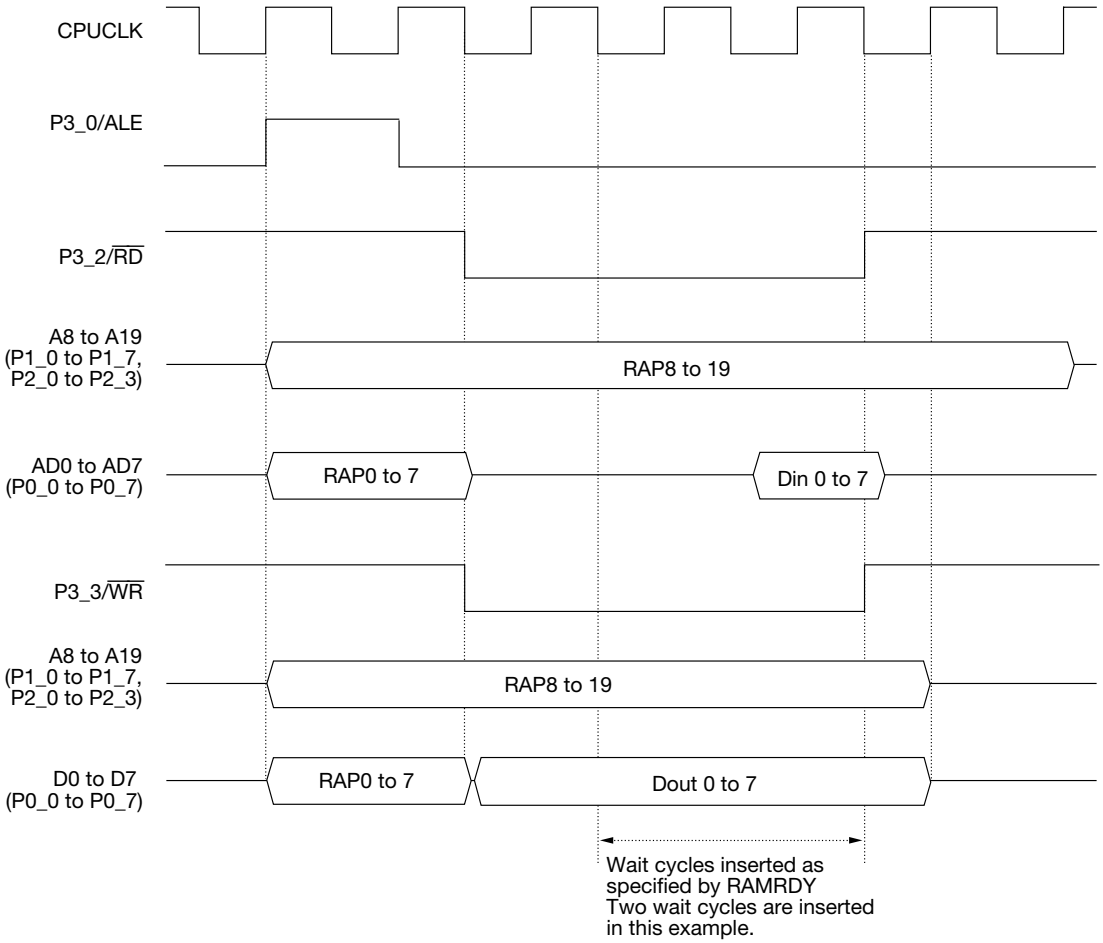


Figure 18-12 External Data Memory Access Timing (Byte Access: 2 Wait Cycles) (Multiplexed Bus Type)

18.5 Notes Regarding Usage of Bus Port Function

18.5.1 Dummy Read Strobe Output

The MSM66577 family of microcontrollers utilize the nX-8/500S, Oki's proprietary 16-bit CPU core.

The instruction code of the nX-8/500S uses 8 bits as its basic unit and consists of 1 to 6 bytes. Instructions are classified as NATIVE instructions for commonly performed operations or as COMPOSIT instructions to realize a wide range of addressing. NATIVE instructions consist of 1 to 4 bytes and are used to achieve high coding and processing efficiency.

COMPOSIT instructions consist of a 1 to 3 byte address field (PREFIX) and a 1 to 3 byte operation field (SUFFIX). The PREFIX and SUFFIX are combined to realize a wide range of addressing.

If instructions accompanying a write to external data memory are to be executed, an unnecessary $\overline{RD}$ signal (dummy $\overline{RD}$) will be output before the actual access ($\overline{WR}$ signal) if some of those instructions are COMPOSIT instructions, (This is limited to cases where the PREFIX specifies an external data memory area.) For byte and bit accesses, a dummy RD signal is output once. For word accesses, a dummy RD signal is output twice.

Some considerations must be exercised in cases where the above mentioned read strobe affects the internal operation of peripheral devices.

Using the bus port function, the specific example of connecting and accessing the 8251 serial interface LSI chip as a peripheral device will be described.

[Example]
When the microcomputer writes to the transmit buffer of the 8251, if COMPOSIT instructions are used with the above conditions, output of the dummy read strobe will cause data in the receive buffer to be read. If receive data exists in the receive buffer, the 8251 will determine that the CPU has finished reading data, and the receive ready output signal will be reset.

Some considerations must be exercised in cases where peripheral devices operate differently when read and write operations are performed at the same address.

These types of problems can be avoided by using NATIVE instructions.

For example, a dummy strobe is not output if load and store instructions (L, LB, ST, STB) are used to read from and write to the accumulator (ACC). (If programming in C language, these sections can be written as assembler functions.)

If general-purpose memory (RAM or ROM) is connected to a bus port, the problems described above should not occur. Problems only occur if a connected peripheral device (functional device) is accessed using COMPOSIT instructions as described above and the read strobe affects the internal operation of the peripheral device.

Connect peripheral devices to bus ports based on an understanding of the operation described herein and the function and operation of peripheral devices.

Tables 18-5 and 18-6 list PREFIX and SUFFIX combinations (instructions) that output a dummy RD when an external data memory area is accessed.

In the tables, PREFIX addressing is inserted at the "*" in the SUFFIX column.

Table 18-5 Instructions (Byte/Bit Manipulations) in Which a Dummy RD Occurs Once (PREFIX and SUFFIX Combinations)

SUFFIX

Instruction symbol	Instruction code
SB *	08+bit
RB *	00+bit
SBR *	B8
RBR *	B9
TBR *	CA
MB *.bit, C	18+bit
MBR *.bit, C	BB
MBR C, *.bit	BA
MOVB *,A	AA
MOVB *,#N8	AB
CLRB *	C7
FILLB *	D7

PREFIX

*	Instruction code
Rn	68+n
[X1]	B0
[DP]	B2
[DP–]	B1
[DP+]	B3
off	B5
dir	B7
N16[X1]	B8
N16[X2]	B9
n7[DP]	9B
n7[USP]	9B
[X1+A]	BA
[X1+R0]	BB

Table 18-6 Instructions (Word Manipulations) in Which a Dummy RD Occurs Twice (PREFIX and SUFFIX Combinations)

SUFFIX

Instruction symbol	Instruction code
MOV *,A	AA
MOV *,#N16	AB
CLR *	C7
FILL *	D7

PREFIX

*	Instruction code
ERn	64+n
[X1]	A0
[DP]	A2
[DP–]	A1
[DP+]	A3
off	A5
dir	A7
N16[X1]	A8
N16[X2]	A9
n7[DP]	8B
n7[USP]	8B
[X1+A]	AA
[X1+R0]	AB

18.5.2 External Bus Access Timing

The MSM66577 family employs a high-speed separate address and data bus type for external bussing.

If the MSM66577 family runs at high speed, the read and write strobe signals may be in their logically active states before the address outputs are changed and then set up. Care must be taken for AC characteristics when peripheral devices, such as general-purpose SRAMs, whose address outputs have to be set up on the rising edge of the $\overline{WR}$ signal are connected using the bus port function.

Refer to the AC characteristics (Chapter 20, "Electrical Characteristics") for operating frequencies to be used.

The AC characteristic values are given under the conditions listed below:

- Load capacitor C_L = 50 pF
- Measurement point for AC timing
  V_{OL}/V_{OH} = 0.8 V/2.0 V for V_{DD} = 4.5 to 5.5 V
  V_{OL}/V_{OH} = $0.16V_{DD}/0.44V_{DD}$ for V_{DD} = 2.4 to 3.6 V
- Ambient temperature Ta = –30 to +70°C
- Power supply voltage V_{DD} = 2.4 to 3.6 V/4.5 to 5.5 V

In order to set up the address value before the $\overline{WR}$ signal is made to its logically active state under the above conditions, either of the following operating frequencies should be provided:

- f = 11 MHz or less for V_{DD} = 2.4 to 3.6 V
- f = 20 MHz or less for V_{DD} = 4.5 to 5.5 V

The upper limits of operating frequencies will actually be varied according to conditions (load capacitance and supply voltages on the printed circuit boards) of products to which the MSM66577 family is to be applied.

The contents described above should be considered to connect the peripheral devices, such as general-purpose SRAMs.

Chapter 19

Flash Memory

19. Flash Memory (Preliminary)

19.1 Overview

The MSM66Q577LY/MSM66Q577, members of the MSM66577 family devices, are equipped with an electrically programmable non-volatile memory (128KB flash memory) as the internal program memory. With three types of flash memory programming modes, the MSM66Q577LY/MSM66Q577 can be programmed even after being installed in a system.

There are two types of flash ROM version; the MSM66Q577 operates in the range of 4.5 to 5.5 V and the MSM66Q577LY operates in the range of 3.0 to 3.6 V. Select the device that matches the voltage range to be used.

19.2 Features

- Power supply voltage

  MSM66Q577 (4.5 to 5.5 V operation) … Can be programmed with a single 4.5 to 5.5 V power supply

  MSM66Q577LY (3.0 to 3.6 V operation) … Can be programmed with a single 3.0 to 3.6 V power supply

- Programming modes

  Flash memory has the following three programming modes.
  - · Parallel mode … Can be programmed with a PROM writer*
  - · Serial mode … Can be programmed with a flash memory writer
  - · User mode … Can be programmed by program execution.

  * Programming operation has been checked by Model 1890A/1893/1930/1931 from MINATO ELECTRONICS, Inc. Flash memory cannot be programmed by other manufacturer's PROM writers.

- Programming blocks

  Flash memory is programmed in blocks of 128-byte units.

- Auto-erase function

  Since an auto-erase function is provided to automatically erase the block to be written to prior to programming, it is unnecessary to erase flash memory before programming.

- Programming Time

  Programming time for the flash memory is listed below.
  - · Programming (128 byte unit) … Approx. 40 ms (at 4.5 to 5.5 V)
    Approx. 50 ms (at 3.0 to 3.6 V)

- Write protect function

  Flash memory has a built-in power-on write protect function that automatically disables programming for approximately 20 ms after power is turned on.

  In addition, there is an acceptor function to prevent incorrect programming in the user mode due to a running of out-of-control program.

- Security function
  Flash memory has a built-in security function that disables reading of contents of memory externally and / or programming externally. The security function is set in the serial mode, but once the security function is set, contents of memory cannot be externally read from or programming cannot be performed externally, in any programming mode.

19.3 Programming Modes

Flash memory of the MSM66Q577LY/MSM66Q577 has the following three programming modes. Since an auto-erase function is provided for all the programming modes, it is not necessary to erase the flash memory prior to programming.

(1) Parallel mode
Programming in this mode is performed with a PROM writer*. A special program to write to flash memory is unnecessary.

In the parallel mode, connect Oki Electronics' flash memory program conversion adapter (model no. MTP66573) and then perform the programming.

* Programming operation has been checked by Model 1890A/1893/1930/1931 from MINATO ELECTRONICS, Inc. Flash memory cannot be programmed by other manufacturer's writers.

(2) Serial mode
Programming in this mode is performed with a flash memory writer. A special program to write to flash memory is unnecessary. Flash memory can be programmed by a single microcontroller or after it is mounted on a printed circuit board.

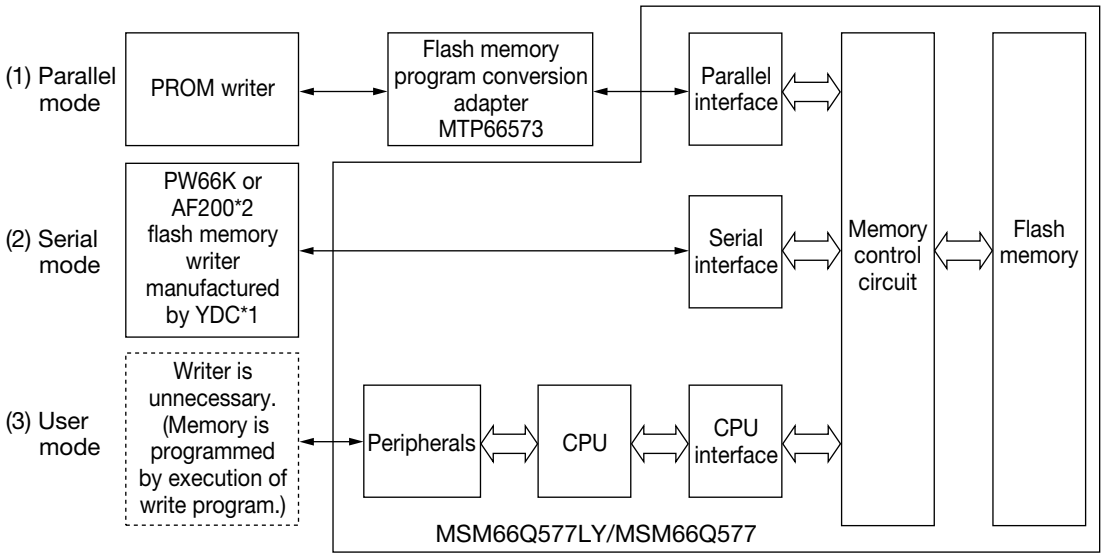
In the serial mode, connect a flash memory writer (model no. PW66K) or a flash microcontroller programmer (model no. AF200) manufactured by Yokogawa Digital Computer Co., Ltd. to the two microcontroller pins (P9_2, P9_3), the $\overline{EA}$ pin, and V_{DD} and GND pins, and then perform the programming. Programming is performed while the microcontroller is in the reset or STOP modes.

(3) User mode

In this mode, instead of using a programming writer, programming is performed by executing a program that writes to flash memory. Programming can be performed after the device is mounted on the circuit board.

In the user mode, programming is performed by executing a write program already stored (using either the serial mode or parallel mode) in flash memory of the microcontroller.

Figure 19-1 shows a block diagram of the flash memory programming modes.



*1: YDC is Yokogawa Digital Computer
*2: AF200 is a trademark of Yokogawa Digital Computer, Co., Ltd.

Figure 19-1 Block Diagram of Programming Modes

19.4 Parallel Mode

19.4.1 Overview of the Parallel Mode

Programming in the parallel mode is performed with a PROM writer*. The writing and reading of programs is performed by connecting Oki Electric's flash memory program conversion adapter (MTP66573) to a PROM writer. Figure 19-2 shows a connection diagram. Since an auto-erase function is provided, flash memory does not have to be erased prior to programming.

* Programming operation has been checked by Model 1890A/1893/1930/1931 from MINATO ELECTRONICS, Inc. Flash memory cannot be programmed by other manufacturer's PROM writers.

19.4.2 PROM Writer Setting

Set the device name to "ATMEL AT29C010A" or "MSM66Q577" using the PROM writer from MINATO ELECTRONICS, Inc.

There are two types of flash ROM version, supporting programming voltages of 5 V and 3 V. In some cases, a custom adapter for the PROM writer may be necessary. Refer to the PROM writer manual for details.

19.4.3 Flash Memory Programming Conversion Adapter

Use Oki Electric's flash memory program conversion adapter (MTP66573).

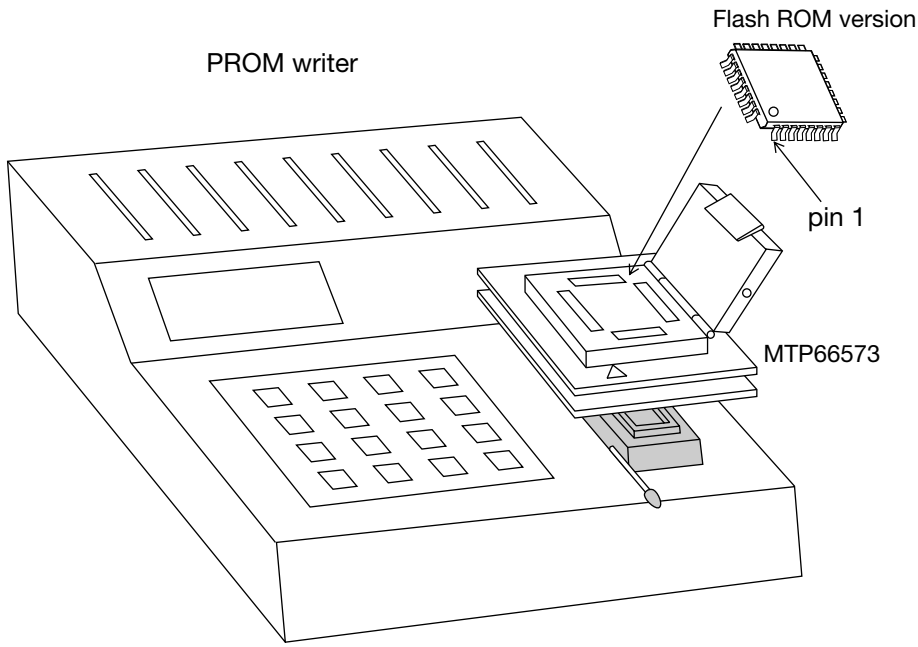


Figure 19-2 Parallel Mode Connection Diagram

19.5 Serial Mode

19.5.1 Overview of the Serial Mode

Programming in the serial mode is performed with a flash memory writer. Programs can be written or read by a single microcontroller or after it is mounted on a printed circuit board.

In the serial mode, the writing and reading of programs is performed by connecting a flash memory writer (PW66K) or a flash microcontroller programmer (AF200) manufactured by Yokogawa Digital Computer, Co., Ltd., to the two microcontroller pins (P9_2, P9_3), the $\overline{EA}$ pin, and V_{DD} and GND pins. Programming and reading are performed while the microcontroller is in the reset or stop modes. Since an auto-erase function is provided, flash memory does not have to be erased prior to programming.

19.5.2 Serial Mode Settings

The serial mode is set automatically by connecting the flash microcontroller programmer to the specific pins and then executing a programming or read operation. When writing or reading is complete, the serial mode setting is released.

(1) Pins used in serial mode

Table 19-1 lists the pins used in the serial mode.

The serial mode can only be set while the CPU is in reset or STOP modes. Be careful of the high voltage (approx. 9 V) that the flash microcontroller programmer applies to the $\overline{EA}$ pin to set the serial mode. V_{DD} is connected to monitor V_{DD} of the user system.

Table 19-1 List of Pins Used in Serial Mode

Pin name	Flash memory function
P9_2	FLACK (serial clock input)
P9_3	FLADAT (serial data I/O)
$\overline{EA}$	FLAMOD (high voltage input to set serial mode)
V_{DD}	User system V_{DD} monitor
GND	Ground

Note1: During the serial mode, the voltage higher than the power supply (approx. 9 V) is applied to the $\overline{EA}$ pin by the flash microcontroller programmer.

Note2: When programming in the serial mode is performed during the STOP mode, set EXINT 6 and 7 valid edges of EXI1CON to interrupt input invalid. (EXI1CON settings, P16-3)

(2) Serial mode connection circuit

In the serial mode, the flash memory writer (PW66K) or the flash microcontroller programmer (AF200) must be connected to the P9_2, P9_3, $\overline{EA}$, V_{DD} and GND pins of the MSM66Q577LY/MSM66Q577 in the user system. In addition, install a switch in the user system to cut off the user system during programming and reading in the serial mode.

Figure 19-3 shows serial mode connection circuit example 1.

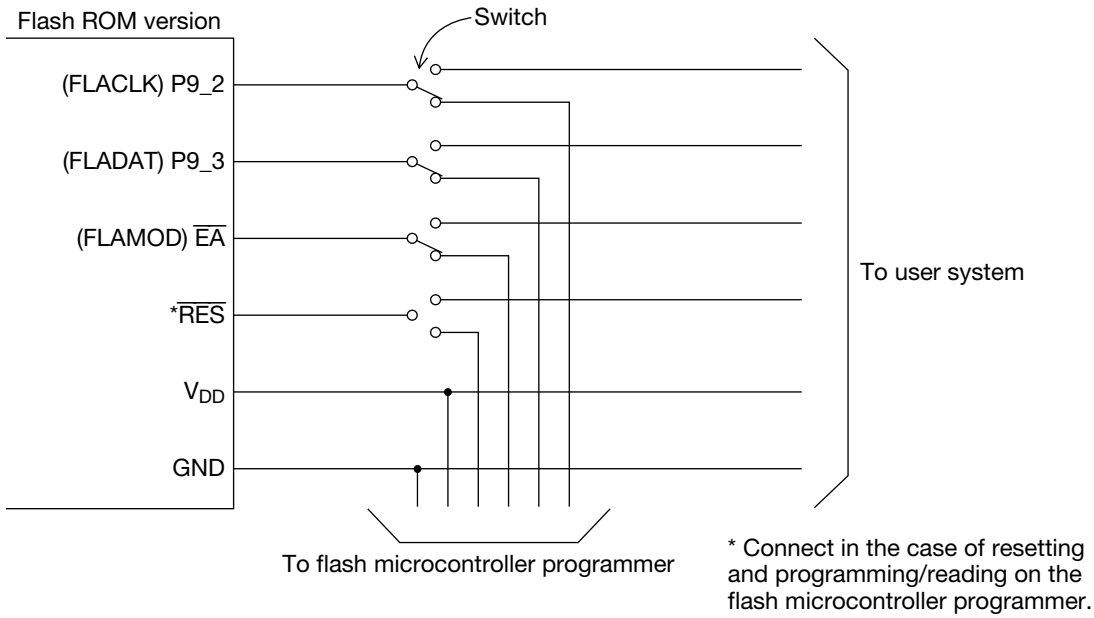


Figure 19-3 Serial Mode Connection Circuit Example 1

If not possible to install a switch in the user system, do not use pins P9_2 and P9_3 with the user system and connect them only to the flash microcontroller programmer. Also, connect each of the P9_2, P9_3 and $\overline{EA}$ pins through a resistor of approximately 100 kΩ to V_{DD}.

Figure 19-4 shows serial mode connection circuit example 2.

Note

The programming and reading in the serial mode are performed while the microcontroller is in the reset or stop mode. To execute the programming/reading during reset, apply "L" level to the $\overline{RES}$ pin. In the case where the flash microcontroller programmer does not apply "L" level to the $\overline{RES}$ pin, the "L" level should be applied by the user application system.

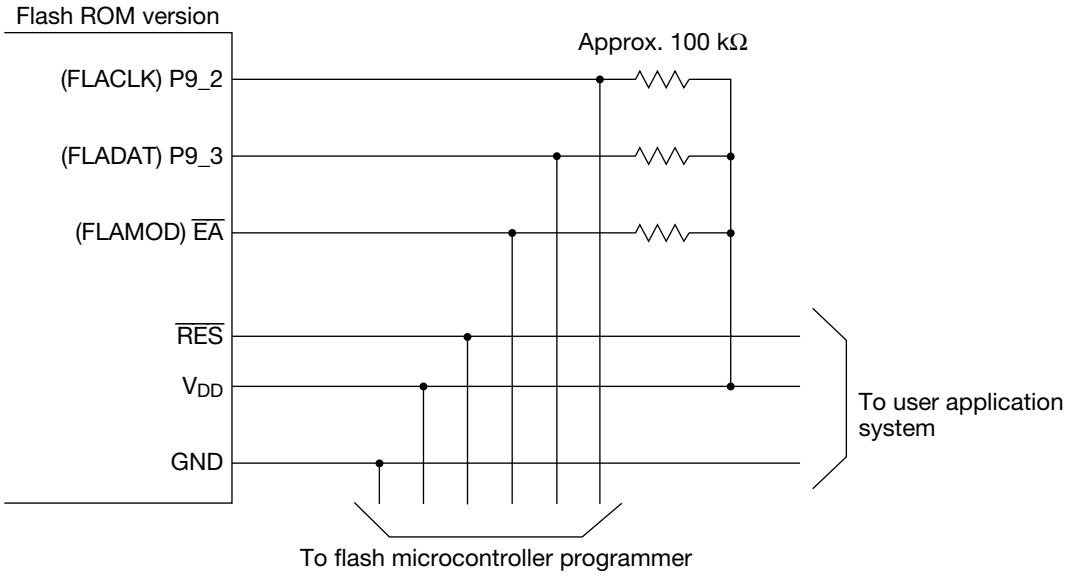


Figure 19-4 Serial Mode Connection Circuit Example 2

(3) Serial mode programming method

Programming in the serial mode is performed with the use of a flash memory writer (PW66K) or a flash microcontroller programmer (AF200) manufactured by Yokogawa Digital Computer.

The procedure for programming with the flash microcontroller programmer is listed below. Refer to the PW66K and AF200 User's Manuals for details of the flash microcontroller programmer.

1) Connect the flash microcontroller programmer to the P9_2, P9_3, $\overline{EA}$, V_{DD} and GND pins of the MSM66Q577LY/MSM66Q577.
2) Set the microcontroller to the reset or STOP mode.
· The flash microcontroller will generate a protocol error if other than reset or STOP modes are set.
3) Perform the programming or read operation with the flash microcontroller programmer.
· The serial mode is set automatically.
4) Verify that operation of the flash microcontroller programmer has been completed correctly.
· The serial mode is released automatically.
5) Release reset or the STOP mode.
· The CPU runs the program that has been written.

(4) Setting of security function

The security function can be set or reset in the serial mode. For the setting method, refer to the User's Manual for the flash microcontroller programmer.
When the security function is set, the flash memory outputs 0s, for external reading, throughout its entire area and programming are disabled, in all programming modes.

(5) Notes on use of serial mode

If programming is performed during the STOP mode, while programming is in progress, do not generate an interrupt or a reset via the $\overline{RES}$ pin input. If generated, the CPU may run out of control after the serial mode is released. During the STOP mode, execution of BRK instructions, overflow of the watchdog timer, and opcode traps will not generate reset. If an interrupt or reset is generated, reprogram the entire flash memory area.

19.6 User Mode

19.6.1 Overview of the User Mode

Instead of using a programming writer, programming in the user mode is performed by executing a program on the user's system to write to flash memory. Programming can be performed even after the microcontroller is mounted on a circuit board in the user's system. Since an auto-erase function is provided, flash memory does not have to be erased prior to programming.

The user mode executes a program to write to flash memory. The program is prepared to contain commands to execute the write operation and the I/O method of data to be written. The program must be written (using either the serial mode or parallel mode) to flash memory in advance.

Figure 19-5 shows a block diagram of the user mode.

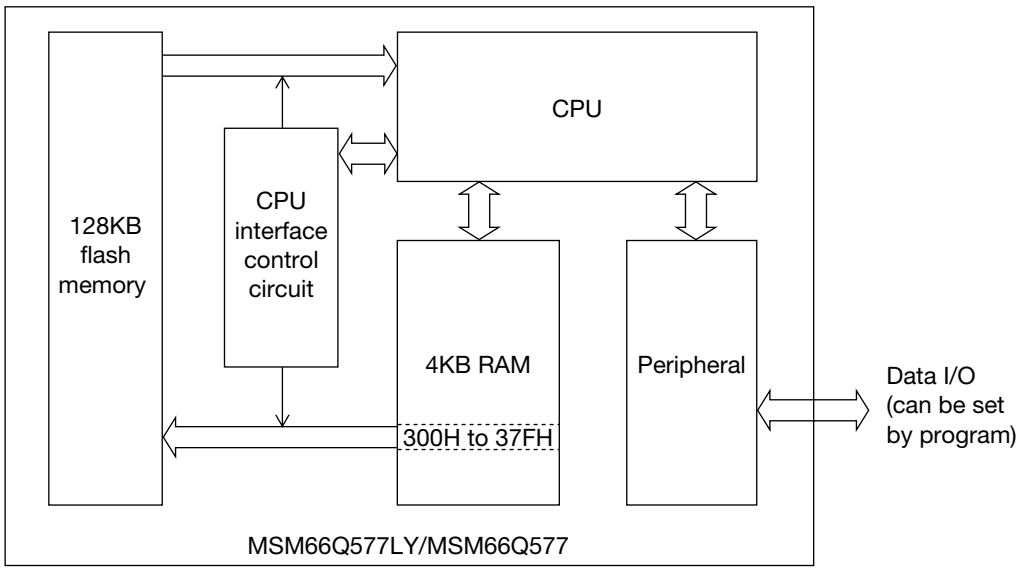


Figure 19-5 User Mode Block Diagram

19.6.2 User Mode Programming Registers

The MSM66Q577LY/MSM66Q577 has internal special function registers (SRFs) for programming with the user mode. Programming in the user mode is performed by controlling the following registers: the flash memory control register (FLACON), the flash memory address register (FLAADRS) and the flash memory acceptor (FLAACP).

Table 19-2 lists a summary of the SFRs for the user mode.

Table 19-2 Summary of SFRs for User Mode

Address [H]	Name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
00F0☆	Flash memory acceptor	FLAACP	—	W	8	"0"	19-11
00F1☆	Flash memory control register	FLACON	—	R/W	8	C6	19-12
00F2☆	Flash memory address register	—	FLAADRS	R/W	16	Undefined	19-11
00F3		—					

Notes:

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 21, "Special Function Registers (SFRs)".

19.6.3 Description of User Mode Registers

(1) Flash memory address register (FLAADRS)

Bits 3 to 12 (FA7 to FA16) of the FLAADRS register set the flash memory address to be programmed.

Figure 19-6 shows the configuration of FLAADRS.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAADRS	—	—	—	FA16	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	FA7	—	—	—
At reset	1	1	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	1	1	1

Address: 00F2 [H]
R/W access: R/W

FA16 to 7									Programming address
15	14	13	12	11	10	9	8	7	
0	0	0	0	0	0	0	0	0	0000H to 007FH
0	0	0	0	0	0	0	0	1	0080H to 00FFH
0	0	0	0	0	0	0	1	0	0100H to 017FH
1	1	1	1	1	1	1	0	1	1FE80H to 1FEFFH
1	1	1	1	1	1	1	1	0	1FF00H to 1FF7FH
1	1	1	1	1	1	1	1	1	1FF80H to 1FFFFH

"—" indicates a nonexistent bit.
When read, its value will be "1".

Figure 19-6 FLAADRS Configuration

(2) Flash memory acceptor (FLAACP)

FLAACP is an acceptor used when data is to be set in the flash memory control register. FLAACP is set to "1" when the program writes n5H, nAH (n = 0 to F) consecutively. Programming the flash memory resets FLAACP to "0".

Figure 19-7 shows the FLAACP configuration.

	7	6	5	4	3	2	1	0
FLAACP								

Address: 00F0 [H]
R/W access: W

Consecutive writing of n5H, nAH enables writing to FLACON.

Figure 19-7 FLAACP Configuration

(3) Flash memory control register (FLACON)

FLACON is a 4-bit register that controls programming and operation of the flash memory.

Figure 19-8 shows the FLACON configuration.

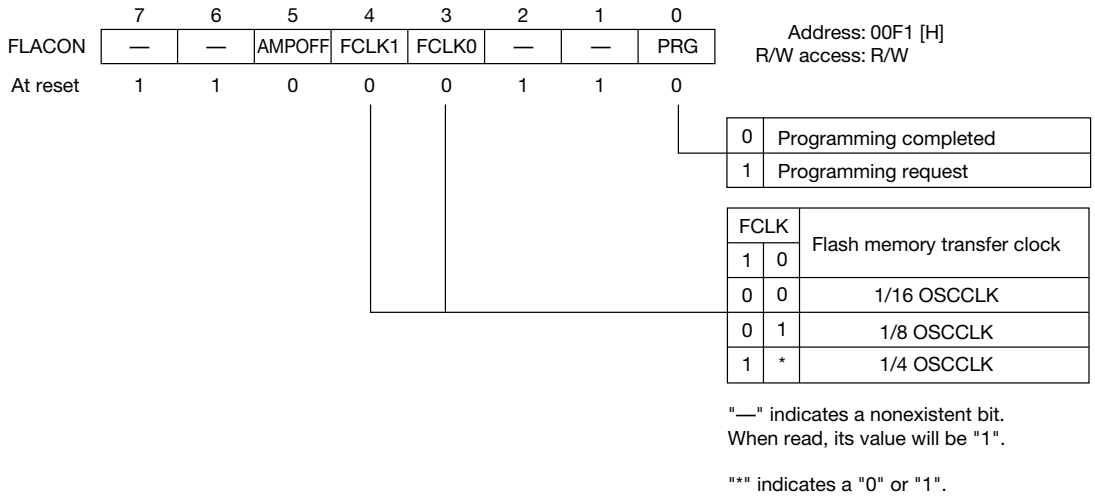


Figure 19-8 FLACON Configuration

Writes to bits 0, 3, and 4 of FLACON are valid after consecutively writing n5H, nAH (n = 0 to F) to FLAACP so that the flash memory acceptor is set to "1". FLAACP is reset to "0" after the flash memory is programmed. To reprogram the flash memory, it is necessary to once again set the flash memory acceptor to "1". Also, while the security is being set, it is not able to write to bits 0, 3, and 4 of the FLACON.

[Description of each bit]

· PRG (bit 0)

If PRG (bit 0) of FLACON is set to "1", after the execution of one instruction, the CPU will enter the hold state and data at internal RAM addresses 300H through 37FH will be transferred to flash memory. Then, a flash memory block will be cleared by the auto-erase function and the write operation performed.

After programming is completed, the hold state is released and this bit is reset to "0". Prior to programming, set the address to be written in FLAADRS and the data to be written in internal RAM addresses 300H to 37FH.

Notes:

· If an interrupt occurs during programming of the flash memory, processing of the interrupt is suspended. The interrupt is processed after programming is completed.

· If reset is initiated by input to the $\overline{\text{RES}}$ pin during programming of the flash memory, the reset is processed after programming is normally completed in the flash memory.

Figure 19-9 shows the relationship between internal RAM and flash memory.

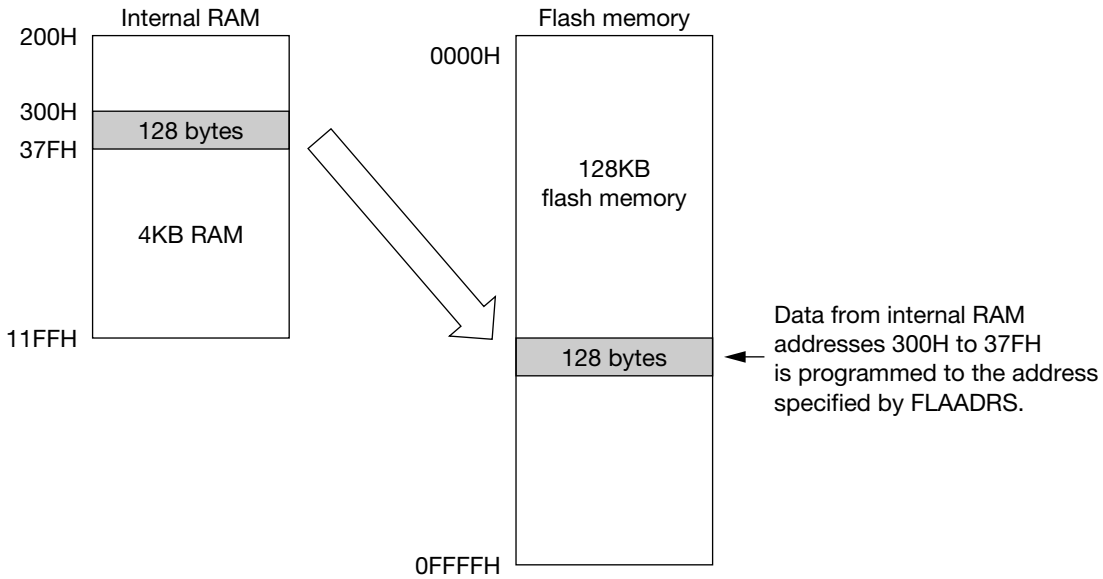


Figure 19-9 Relation between Internal RAM and Flash Memory

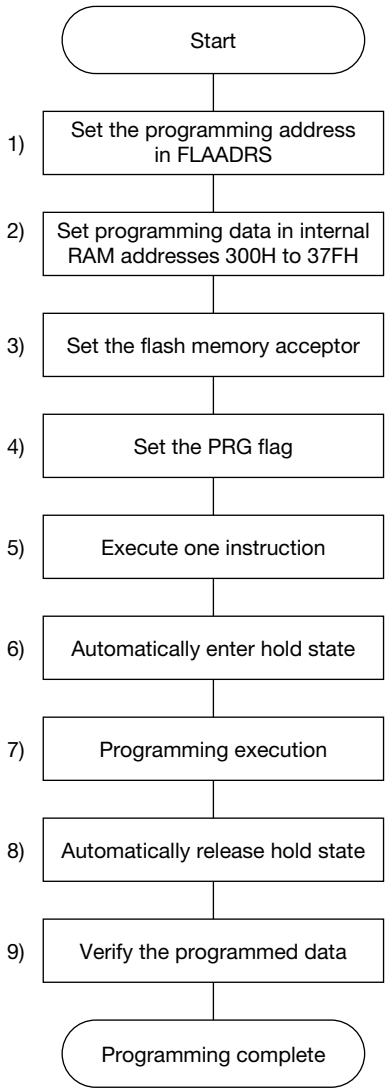
· FCLK0, FCLK1 (bits 3,4)
FCLK0 (bit 3) and FCLK1 (bit 4) of the FLACON register are bits that set the clock that transfers data from internal RAM addresses 300H through 37FH to flash memory. At reset, since both FCLK0 and FCLK1 become "0", 1/16CLK will be selected as the transfer clock.
Set FCLK0 and FCKL1 such that the transfer clock frequency is 10 MHz or less for the MSM66Q577 (4.5 to 5.5 V programming voltage) and 6.6 MHz or less for the MSM66Q577LY (3.0 to 3.6 V programming voltage).

· AMPOFF (bit 5)
AMPOFF (bit 5) of FLACON is a bit that is provided to control the sense amplifiers of the flash memory. However, the sense amplifiers constantly have intermittent access to the flash memory regardless of the state of this bit, in which case the flash memory enters the low power mode.

19.6.4 User Mode Programming Example

(1) User mode programming flowchart example

Figure 19-10 shows a flowchart for user mode programming of flash memory.
Since an auto-erase function is prepared, flash memory does not have to be erased prior to programming.



1) Set FLAADRS (FA7 to FA16) with the upper ten bits of the address desired to write to in 128 byte units.

2) Using serial I/O or external memory access, set the programming data in internal RAM addresses 300H through 37FH.

3) Write the consecutive values of n5H, nAH (n = 0 to F) to FLAACP to set the flash memory acceptor to "1." Programming the flash memory causes the flash memory acceptor to be reset to "0."

4) Set the program (PRG) flag of FLACON. Programming causes the PRG flag to be reset to "0."

5) After the PRG flag is set, the CPU executes one instruction. (Example: NOP instruction)

6) After the execution of one instruction, the CPU automatically enters the hold state.

7) When the CPU enters the hold state, data in internal RAM addresses 300H through 37FH is transferred to the flash memory, and programming (with the auto-erase function) is performed.

8) After completion of flash memory programming, the hold state is automatically released and the CPU executes the next instruction.

9) Verify that data has been written to the flash memory. ROM table reference instructions may be used to verify the data.

Note: If programming has been performed by an external program, the programmed data cannot be verified.

Figure 19-10 Programming Flowchart Example

(2) User mode programming program example

Listed below is an example program that programs data to flash memory addresses 5500H through 557FH (128 bytes) and then verifies the data of those 128 bytes.

It is assumed that the data to be programed has already been stored in internal RAM addresses 300H through 37FH.

```
        MOV     FLAADRS,#0550H      Set the start address (5500H) for programming.
        MOVB    FLAACP,#05H
        MOVB    FLAACP,#0AH         Set the flash memory acceptor.
        MOVB    FLACON,#11H         Set the PRG flag.
        NOP                         After execution of one instruction, the hold state
                                    is entered and programming begins. When
                                    programming is complete, the hold state is
                                    released.

        MOV     DP,#300H            Set the internal RAM address in DP.
        MOV     ER0,#5500H          Set the flash memory address in ER0.
        SDD                         Set the data descriptor.

LOOP:

        LC      A, [ER0]            Load flash memory data into the accumulator.
        CMP     A, [DP+]            Compare accumulator and internal RAM data,
                                    then increment the internal RAM address by +2.
        JC      NE,ERR              If they are not equal, jump to the error routine.
        ADDB    R0,#02H             Increment the flash memory address by +2.
        JBR     R0.7,LOOP           If verification of the 128 bytes is not complete,
                                    jump to LOOP.

        ERR:                        Perform error processing.
```

Note: If programming has been performed by an external program, the programmed data cannot be verified.

19.6.5 Notes on Use of User Mode

Note the following items when generating a program to be used with the user mode.

- If an interrupt occurs during programming of the flash memory, processing of the interrupt is put on hold. The interrupt is processed after programming is completed.
- If reset is initiated by input to the $\overline{\text{RES}}$ pin during programming of the flash memory, the reset is processed. However, the flash memory area that was in the process of being programmed will have been incorrectly programmed. If reset is initiated during programming, reprogram the flash memory area that was in the process of being programmed.
- Do not program to the flash memory area that contains the programming program being executing. (After programming is completed, the CPU program control will run out of control.)
- Development tools (emulator) cannot evaluate programming or erasing.

19.7 Notes on Program

(1) Programming of flash memory immediately after power-on

Programming to flash memory is automatically disabled for approximately 20 ms (for both 4.5 to 5.5 V and 3.0 to 3.6 V devices) after power is turned on. Therefore, if flash memory is to be programmed immediately after power is turned on, wait for the above time by guaranteeing a power-on reset time.

(2) Note on STOP mode release

Flash memory requires a standby time of at least 50 µs when the STOP mode is released. Therefore, set the standby control register (SBYCON) to guarantee the stabilization time for main clock (OSCCLK) oscillation when the STOP is released.

(3) Supply voltage sense reset function

If the internal ROM (high level input to the $\overline{EA}$ pin) of the MSM66Q577LY/MSM66Q577 is operative, the reset function is implemented at a supply voltage of 3 V or less for the MSM66Q577 (the version operating in the range of 4.5 to 5.5 V) and at a supply voltage of 1.5 V or less for the MSM66Q577LY (the version operating in the range of 3.0 to 3.6 V). Programming is automatically disabled for approximately 20 ms after the reset function is implemented. Also, the reset function is not implemented during the STOP mode (only when oscillation of the main clock is terminated).

Chapter 20

Electrical Characteristics

20. Electrical Characteristics (Preliminary)

20.1 Absolute Maximum Ratings

Parameter	Symbol	Condition		Rated value	Unit
Digital power supply voltage	V_{DD}	GND = AGND = 0 V, Ta = 25°C	MSM66577/Q577	–0.3 to +7.0	V
			MSM66577L/Q577LY	–0.3 to +4.6	V
Input voltage	V_I		—	–0.3 to V_{DD} + 0.3	V
Output voltage	V_O		—	–0.3 to V_{DD} + 0.3	V
Analog reference voltage	V_{REF}		—	–0.3 to V_{DD} + 0.3	V
Analog input voltage	V_{AI}		—	–0.3 to V_{REF}	V
Power dissipation	P_D	Ta = 70°C per package	100-pin TQFP	650	mW
Storage temperature	T_{STG}	—		–50 to +150	°C

20.2 Recommended Operating Conditions

Parameter	Symbol	Condition		Range	Unit
Digital power supply voltage	V_{DD}	MSM66577	$f_{OSC} \leq 30$ MHz	4.5 to 5.5	V
		MSM66577L	$f_{OSC} \leq 14$ MHz	2.4 to 3.6	
		MSM66Q577	$f_{OSC} \leq 30$ MHz	4.5 to 5.5	
		MSM66Q577LY	$f_{OSC} \leq 14$ MHz	3.0 to 3.6	
Analog reference voltage	V_{REF}	—		V_{DD} – 0.3 to V_{DD}	V
Analog input voltage	V_{AI}	—		AGND to V_{REF}	V
Memory hold voltage	V_{DDH}	MSM66577/Q577	f_{OSC} = 0 Hz	2.0 to 5.5	V
		MSM66577L/Q577LY		2.0 to 3.6	
Operating frequency	f_{OSC}	MSM66577	V_{DD} = 4.5 to 5.5 V	2 to 30	MHz
		MSM66577L	V_{DD} = 2.4 to 3.6 V	2 to 14	
		MSM66Q577	V_{DD} = 4.5 to 5.5 V	2 to 30	
		MSM66Q577LY	V_{DD} = 3.0 to 3.6 V	2 to 14	
	f_{XT}	—		32.768	kHz
Ambient temperature	Ta	—		–30 to +70	°C
Fan out	N	MOS load		20	—
		TTL load	P0, P3, P11	6	—
			P1, P2, P4, P5, P6, P7, P8, P9, P10, P14, P15	1	—

20.3 Allowable Output Current Values

MSM66577L/577 (V_{DD} = 2.4 to 3.6 V/4.5 to 5.5 V, Ta = –30 to +70°C)
MSM66Q577LY/Q577 (V_{DD} = 3.0 to 3.6 V/4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Pin	Symbol	Min.	Typ.	Max.	Unit
"H" output pin (1 pin)	All output pins	I_{OH}	—	—	–2	mA
"H" output pins (sum total)	Sum total of all output pins	ΣI_{OH}	—	—	–40	
"L" output pin (1 pin)	P0, P3, P11	I_{OL}	—	—	10	
	Other ports				5	
"L" output pins (sum total)	Sum total of P0, P3, P11	ΣI_{OL}	—	—	80	
	Sum total of P1, P2, P4				50	
	Sum total of P5, P6, P9					
	Sum total of P7, P8, P10, P14, P15					
	Sum total of all output pins				140	

[Note]
Each of the family devices has unique pattern routes for the internal power and ground. Connect the power supply voltage to all V_{DD} pins and the ground potential to all GND pins. If a device may have one or more V_{DD} or GND pins to which the power supply voltage or the ground potential is not connected, it cannot be guaranteed for normal operation.

20.4 Internal Flash ROM Programming Conditions

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}	MSM66Q577	4.5 to 5.5	V
		MSM66Q577LY	3.0 to 3.6	V
Ambient temperature	Ta	During Read	–30 to +70	°C
		During Programming	0 to +50	°C
Endurance	CEP	—	100	Cycles
Blocks size	—	—	128	bytes

20.5 DC Characteristics

20.5.1 DC Characteristics (V_{DD} = 4.5 to 5.5 V)

MSM66577/Q577 (V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" input voltage *1	V_{IH}	—	$0.44V_{DD}$	—	V_{DD} + 0.3	V
"H" input voltage *2,*3,*4,*5,*6			$0.80V_{DD}$	—	V_{DD} + 0.3	
"L" input voltage *1	V_{IL}	—	–0.3	—	$0.16V_{DD}$	
"L" input voltage *2,*3,*4,*5,*6			–0.3	—	$0.20V_{DD}$	
"H" output voltage *1, *4	V_{OH}	I_O = –400 µA	V_{DD} – 0.4	—	—	
		I_O = –2.0 mA	V_{DD} – 0.6	—	—	
"H" output voltage *2		I_O = –200 µA	V_{DD} – 0.4	—	—	
		I_O = –1.0 mA	V_{DD} – 0.6	—	—	
"L" output voltage *1, *4	V_{OL}	I_O = 3.2 mA	—	—	0.4	
		I_O = 10.0 mA	—	—	0.8	
"L" output voltage *2		I_O = 1.6 mA	—	—	0.4	
		I_O = 5.0 mA	—	—	0.8	
Input leakage current *3	I_{IH}/I_{IL}	$V_I = V_{DD}/0$ V	—	—	1/–1	µA
Input current *5			—	—	1/–250	
Input current *6			—	—	15/–15	
Output leakage current *1,*2,*4	I_{LO}	$V_O = V_{DD}/0$ V	—	—	±10	µA
Pull-up resistance	R_{pull}	V_I = 0 V	25	50	100	kΩ
Input capacitance	C_I	f_{OSC} = 1 MHz, Ta = 25°C	—	5	—	pF
Output capacitance	C_O		—	7	—	
Analog reference supply current	I_{REF}	During A/D operation	—	—	4	mA
		When A/D is stopped	—	—	10	µA

*1: Applicable to P0
*2: Applicable to P1, P2, P4, P5, P6, P7, P8, P9, P10, P14, P15
*3: Applicable to P12, SELMBUS, $\overline{EA}$, NMI
*4: Applicable to P3, P11
*5: Applicable to $\overline{RES}$
*6: Applicable to OSC0

Supply current (V_{DD} = 4.5 to 5.5 V)

MSM66577/Q577 (V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Mode	Symbol	Condition		Min.	Typ.	Max.	Unit
CPU operation mode *1	I_{DD}	f_{OSC} = 30 MHz		—	60	80	mA
		f_{XT} = 32.768 kHz		—	80	180	µA
HALT mode *2	I_{DDH}	f_{OSC} = 30 MHz		—	40	60	mA
STOP mode *3	I_{DDS}	OSC is stopped	XT is used	—	5	110	µA
			XT is not used	—	1	100	
		OSC is stopped, XT is not used V_{DD} = 2 V, Ta = 25°C		—	0.2	10	

[Note]
Ports used as inputs are at V_{DD} or 0 V. Other ports are unloaded.
*1: CPU and all the peripheral functions (timer, PWM, A/D, etc.) are activated.
*2: CPU is stopped, and all the peripheral functions (timer, PWM, A/D, etc.) are activated.
*3: CPU and all the peripheral functions are deactivated (The clock timer is being activated when the XT is used).

20.5.2 DC Characteristics (V_{DD} = 2.4 to 3.6 V)

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" input voltage *1	V_{IH}	—	$0.55V_{DD}$	—	V_{DD} + 0.3	V
"H" input voltage *2,*3,*4,*5,*6			$0.80V_{DD}$	—	V_{DD} + 0.3	
"L" input voltage *1	V_{IL}	—	–0.3	—	$0.16V_{DD}$	
"L" input voltage *2,*3,*4,*5,*6			–0.3	—	$0.20V_{DD}$	
"H" output voltage *1, *4	V_{OH}	I_O = –400 µA	V_{DD} – 0.4	—	—	
		I_O = –2.0 mA	V_{DD} – 0.8	—	—	
"H" output voltage *2		I_O = –200 µA	V_{DD} – 0.4	—	—	
		I_O = –1.0 mA	V_{DD} – 0.8	—	—	
"L" output voltage *1, *4	V_{OL}	I_O = 3.2 mA	—	—	0.5	
		I_O = 5.0 mA	—	—	0.9	
"L" output voltage *2		I_O = 1.6 mA	—	—	0.5	
		I_O = 2.5 mA	—	—	0.9	
Input leakage current *3,*6	I_{IH}/I_{IL}	$V_I = V_{DD}/0$ V	—	—	1/–1	µA
Input current *5			—	—	1/–250	
Input current *6			—	—	15/–15	
Output leakage current *1,*2,*4	I_{LO}	$V_O = V_{DD}/0$ V	—	—	±10	µA
Pull-up resistance	R_{pull}	V_I = 0 V	40	100	200	kΩ
Input capacitance	C_I	f_{OSC} = 1 MHz, Ta = 25°C	—	5	—	pF
Output capacitance	C_O		—	7	—	
Analog reference supply current	I_{REF}	During A/D operation	—	—	2	mA
		When A/D is stopped	—	—	5	µA

*1: Applicable to P0
*2: Applicable to P1, P2, P4, P5, P6, P7, P8, P9, P10, P14, P15
*3: Applicable to P12, SELMBUS, $\overline{EA}$, NMI
*4: Applicable to P3, P11
*5: Applicable to $\overline{RES}$
*6: Applicable to OSC0

Supply current (V_{DD} = 2.4 to 3.6 V)

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Mode	Symbol	Condition		Min.	Typ.	Max.	Unit
CPU operation mode *1	I_{DD}	f_{OSC} = 14 MHz		—	15	25	mA
		f_{XT} = 32.768 kHz		—	50	150	µA
HALT mode *2	I_{DDH}	f_{OSC} = 14 MHz		—	10	17	mA
STOP mode *3	I_{DDS}	OSC is stopped	XT is used	—	3	110	µA
			XT is not used	—	1	100	
		OSC is stopped, XT is not used V_{DD} = 2 V, Ta = 25°C		—	0.2	10	

[Note]
Ports used as inputs are at V_{DD} or 0 V. Other ports are unloaded.
*1: CPU and all the peripheral functions (timer, PWM, A/D, etc.) are activated.
*2: CPU is stopped, and all the peripheral functions (timer, PWM, A/D, etc.) are activated.
*3: CPU and all the peripheral functions are deactivated (The clock timer is being activated when the XT is used).

20.6 AC Characteristics

20.6.1 AC Characteristics (V_{DD} = 4.5 to 5.5 V)

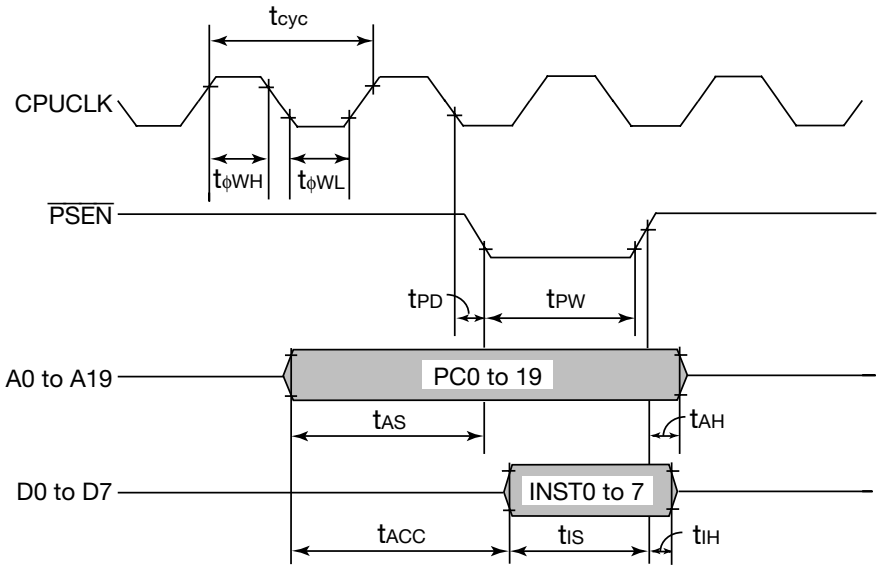
(1) Separate Bus Type

1. External program memory control

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	13	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		13	—	
$\overline{PSEN}$ pulse width	t_{PW}		2 tϕ – 15	—	
$\overline{PSEN}$ pulse delay time	t_{PD}		—	45	
Address setup time	t_{AS}		tϕ – 25	—	
Address hold time	t_{AH}		0	—	
Instruction setup time	t_{IS}		25	—	
Instruction hold time	t_{IH}		0	—	
Read data access time	t_{ACC}		—	3 tϕ – 65*1	

Note: $t\phi = t_{cyc}/2$



Bus timing during no wait cycle time

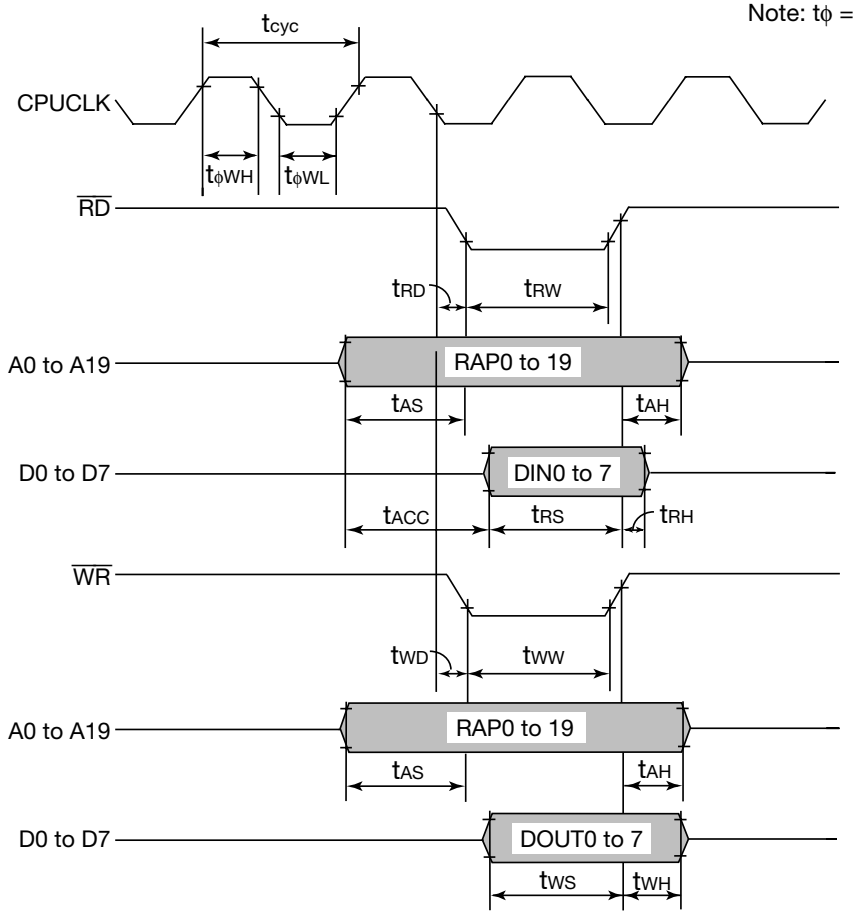
*1 The read data access time (t_{ACC}) is (3 + 2n) tϕ – 65 when n wait cycles are inserted. For more details, refer to Section 18.4, "External Memory Access Timing".

2. External data memory control

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	13	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		13	—	
$\overline{RD}$ pulse width	t_{RW}		2 tϕ – 15	—	
$\overline{WR}$ pulse width	t_{WW}		2 tϕ – 15	—	
$\overline{RD}$ pulse delay time	t_{RD}		—	45	
$\overline{WR}$ pulse delay time	t_{WD}		—	45	
Address setup time	t_{AS}		tϕ – 25	—	
Address hold time	t_{AH}		tϕ – 3	—	
Read data setup time	t_{RS}		25	—	
Read data hold time	t_{RH}		0	—	
Read data access time	t_{ACC}		—	3 tϕ – 65*1	
Write data setup time	t_{WS}		2 tϕ – 30	—	
Write data hold time	t_{WH}		tϕ – 3	—	

Note: $t\phi = t_{cyc}/2$



Bus timing during no wait cycle time

*1 The read data access time (t_{ACC}) is (3 + 2n) tϕ – 65 when n wait cycles are inserted. For more details, refer to Section 18.4, "External Memory Access Timing".

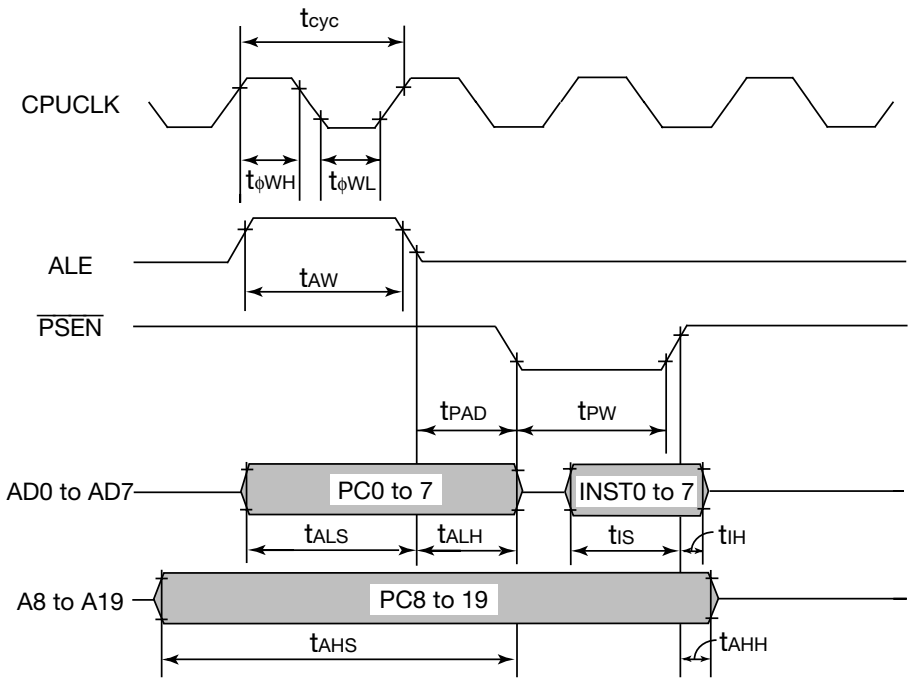
(2) Multiplexed Bus Type

1. External program memory control

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	13	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		13	—	
ALE pulse width	t_{AW}		2 tϕ – 10	—	
$\overline{PSEN}$ pulse width	t_{PW}		2 tϕ – 15	—	
$\overline{PSEN}$ pulse delay time	t_{PAD}		tϕ – 3	—	
Low address setup time	t_{ALS}		2 tϕ – 15	—	
Low address hold time	t_{ALH}		tϕ – 3	—	
High address setup time	t_{AHS}		3 tϕ – 25	—	
High address hold time	t_{AHH}		0	—	
Instruction setup time	t_{IS}		25	—	
Instruction hold time	t_{IH}		0	tϕ – 3	

Note: $t\phi = t_{cyc}/2$



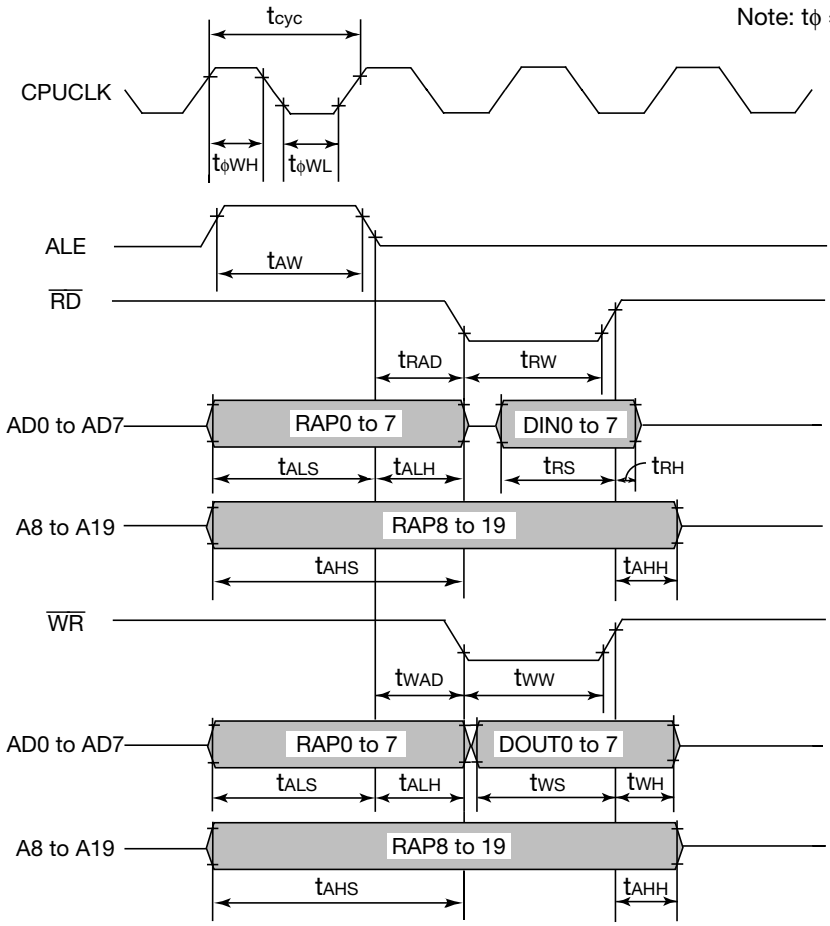
Bus timing during no wait cycle time

2. External data memory control

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70˚C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	13	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		13	—	
ALE pulse width	t_{AW}		2 tϕ – 10	—	
$\overline{RD}$ pulse width	t_{RW}		2 tϕ – 15	—	
$\overline{WR}$ pulse width	t_{WW}		2 tϕ – 15	—	
$\overline{RD}$ pulse delay time	t_{RAD}		tϕ – 3	—	
$\overline{WR}$ pulse delay time	t_{WAD}		tϕ – 3	—	
Low address setup time	t_{ALS}		2 tϕ – 15	—	
Low address hold time	t_{ALH}		tϕ – 3	—	
High address setup time	t_{AHS}		3 tϕ – 25	—	
High address hold time	t_{AHH}		tϕ – 3	—	
Read data setup time	t_{RS}		25	—	
Read data hold time	t_{RH}		0	tϕ – 3	
Write data setup time	t_{WS}		2 tϕ – 30	—	
Write data hold time	t_{WH}		tϕ – 3	—	

Note: $t\phi = t_{cyc}/2$



Bus timing during no wait cycle time

(3) Serial port control

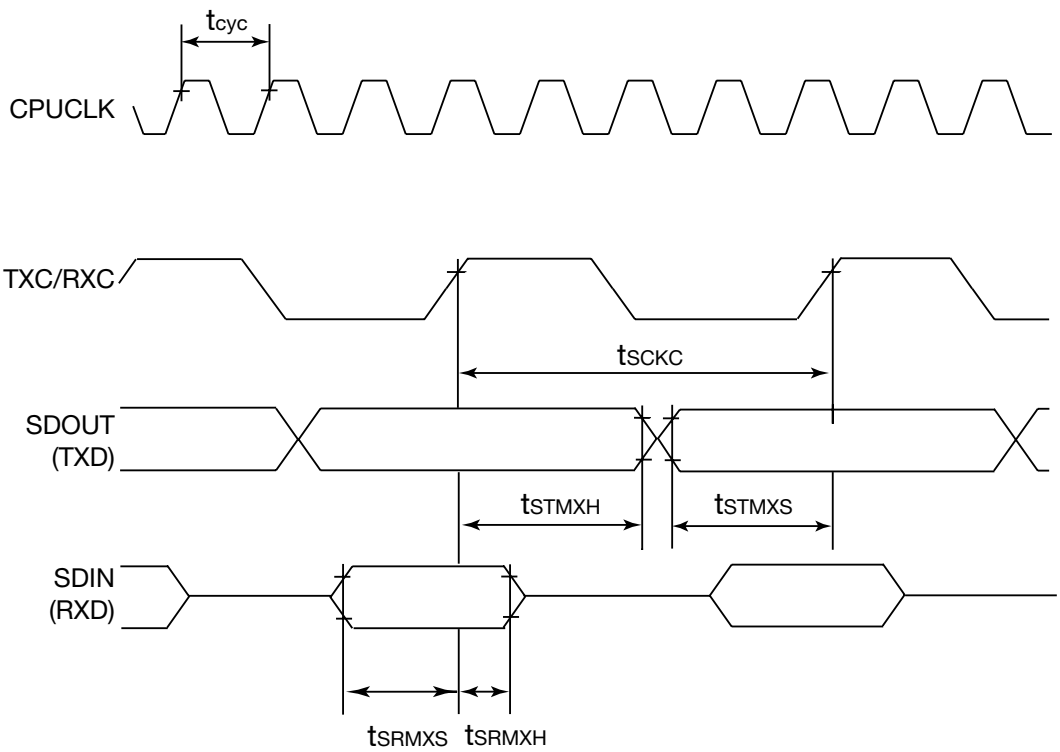
• Serial ports 1 and 6 (SIO1 and 6)

Master mode (Clock synchronous serial port)

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	4 t_{cyc}	—	
Output data setup time	t_{STMXS}		2 tϕ – 5	—	
Output data hold time	t_{STMXH}		5 tϕ – 10	—	
Input data setup time	t_{SRMXS}		13	—	
Input data hold time	t_{SRMXH}		0	—	

Note: $t\phi = t_{cyc}/2$

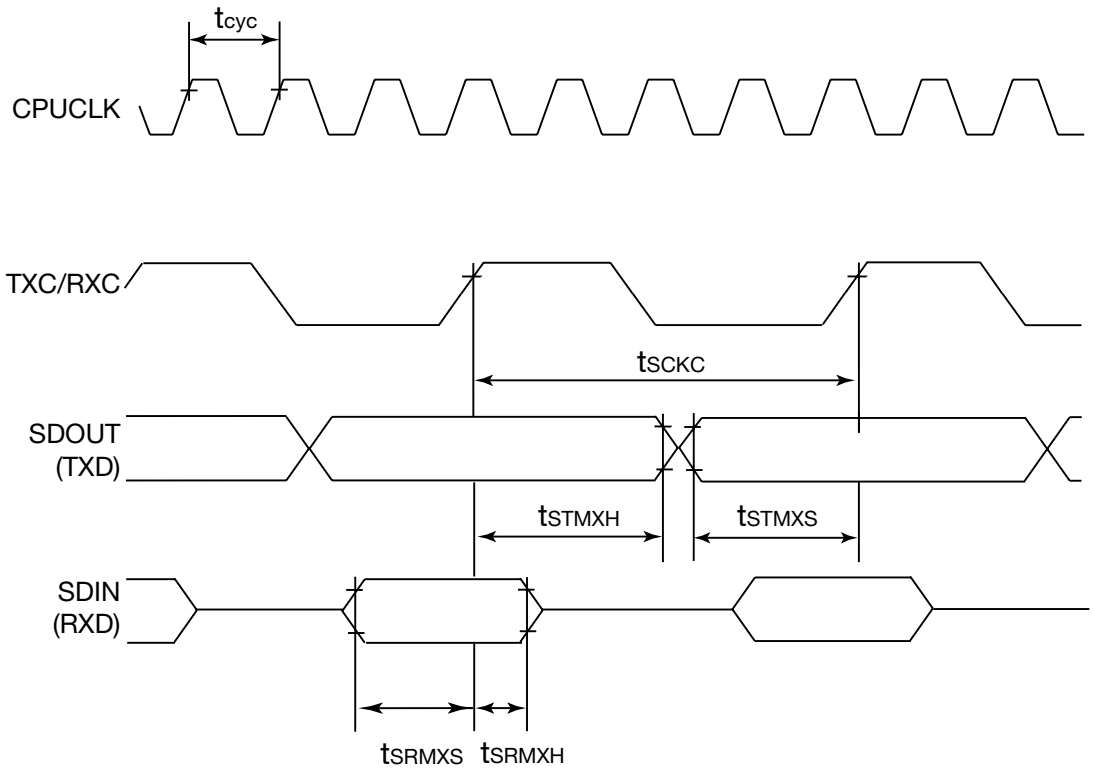


Slave mode (Clock synchronous serial port)

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	4 t_{cyc}	—	
Output data setup time	t_{STMXS}		2 $t\phi$ – 15	—	
Output data hold time	t_{STMXH}		4 $t\phi$ – 10	—	
Input data setup time	t_{SRMXS}		13	—	
Input data hold time	t_{SRMXH}		3	—	

Note: $t\phi = t_{cyc}/2$



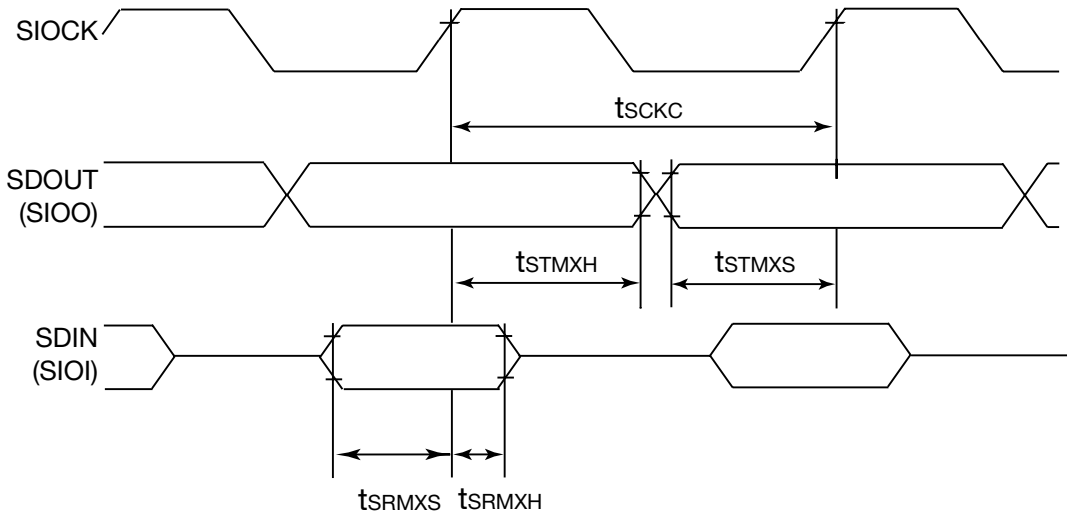
• Serial ports 4 and 5 (SIO4 and 5)

Master mode (Clock synchronous serial port)

(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	6 t_{cyc}	—	
Output data setup time	t_{STMXS}		6 $t\phi$ – 5	—	
Output data hold time	t_{STMXH}		4.5 $t\phi$ – 10	—	
Input data setup time	t_{SRMXS}		13	—	
Input data hold time	t_{SRMXH}		0	—	

Note: $t\phi = t_{cyc}/2$

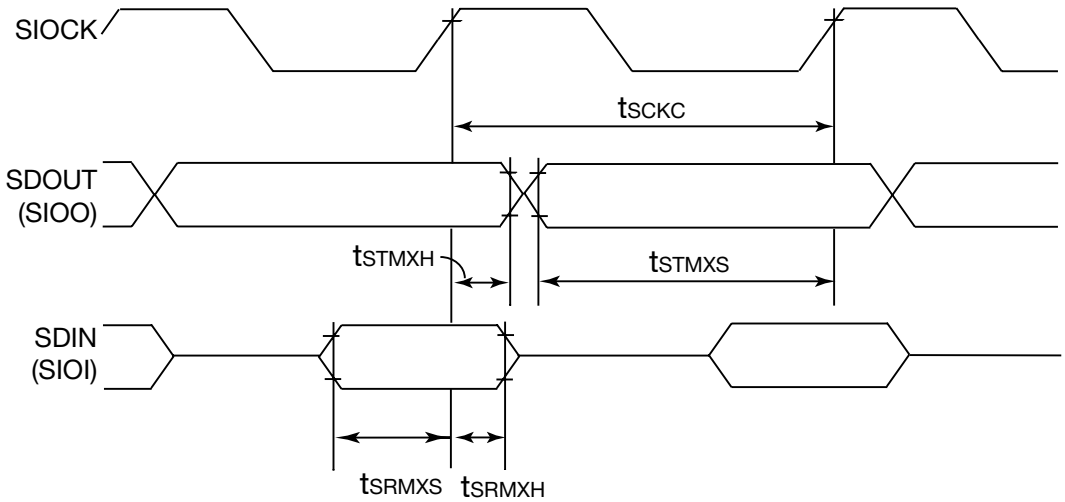


Slave mode (Clock synchronous serial port)

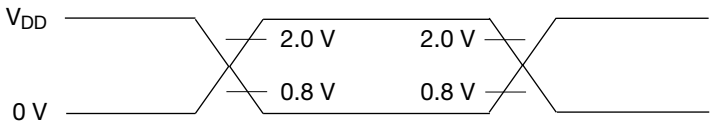
(V_{DD} = 4.5 to 5.5 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 30 MHz	33.3	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	6 t_{cyc}	—	
Output data setup time	t_{STMXS}		3 $t\phi$ – 15	—	
Output data hold time	t_{STMXH}		6 $t\phi$ – 10	—	
Input data setup time	t_{SRMXS}		13	—	
Input data hold time	t_{SRMXH}		3	—	

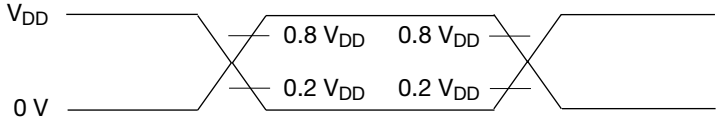
Note: $t\phi = t_{cyc}/2$



Measurement points for AC timing (except the serial port)



Measurement points for AC timing (the serial port)



20.6.2 AC Characteristics (V_{DD} = 2.4 to 3.6 V)

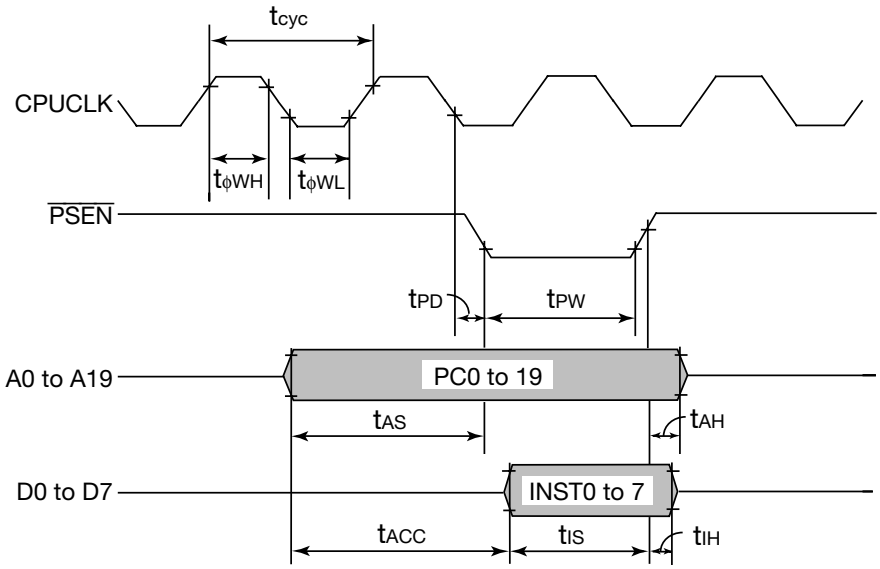
(1) Separate Bus Type

1. External program memory control

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	28	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		28	—	
$\overline{PSEN}$ pulse width	t_{PW}		2 tϕ – 40	—	
$\overline{PSEN}$ pulse delay time	t_{PD}		—	95	
Address setup time	t_{AS}		tϕ – 45	—	
Address hold time	t_{AH}		0	—	
Instruction setup time	t_{IS}		75	—	
Instruction hold time	t_{IH}		0	—	
Read data access time	t_{ACC}		—	3 tϕ – 120*1	

Note: tϕ = t_{cyc}/2



Bus timing during no wait cycle time

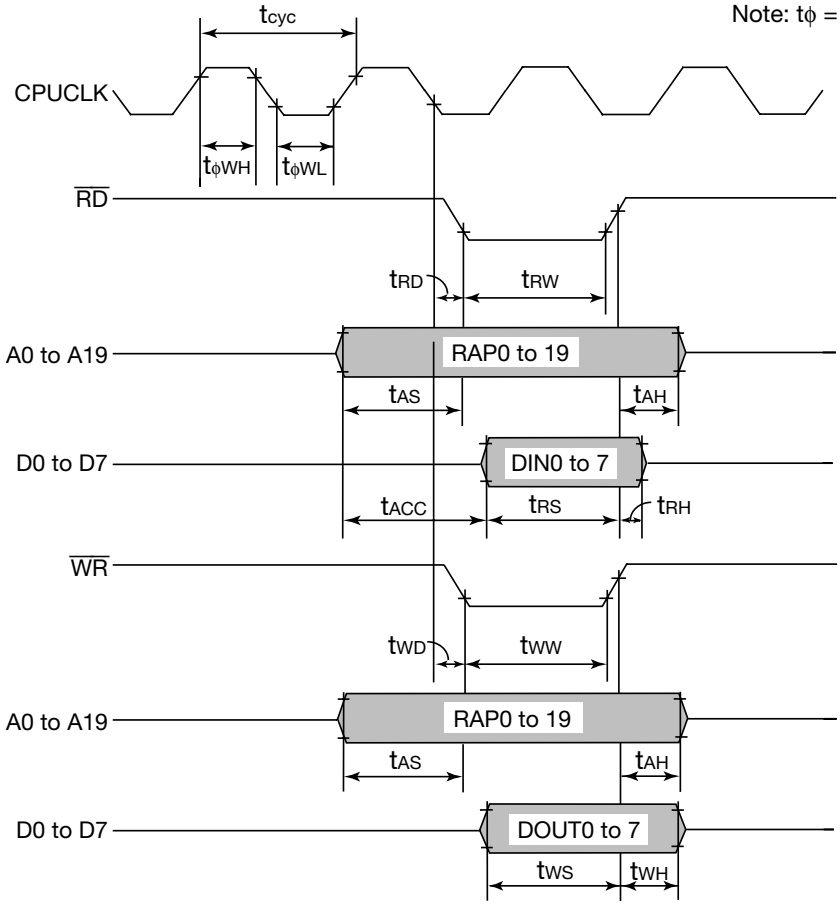
*1 The read data access time (t_{ACC}) is (3 + 2n) tϕ – 120 when n wait cycles are inserted. For more details, refer to Section 18.4, "External Memory Access Timing".

2. External data memory control

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	28	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		28	—	
$\overline{RD}$ pulse width	t_{RW}		2 tϕ – 40	—	
$\overline{WR}$ pulse width	t_{WW}		2 tϕ – 40	—	
$\overline{RD}$ pulse delay time	t_{RD}		—	95	
$\overline{WR}$ pulse delay time	t_{WD}		—	95	
Address setup time	t_{AS}		tϕ – 45	—	
Address hold time	t_{AH}		tϕ – 6	—	
Read data setup time	t_{RS}		75	—	
Read data hold time	t_{RH}		0	—	
Read data access time	t_{ACC}		—	3 tϕ – 120*1	
Write data setup time	t_{WS}		2 tϕ – 55	—	
Write data hold time	t_{WH}		tϕ – 6	—	

Note: tϕ = t_{cyc}/2



Bus timing during no wait cycle time

*1 The read data access time (t_{ACC}) is (3 + 2n) tϕ – 120 when n wait cycles are inserted. For more details, refer to Section 18.4, "External Memory Access Timing".

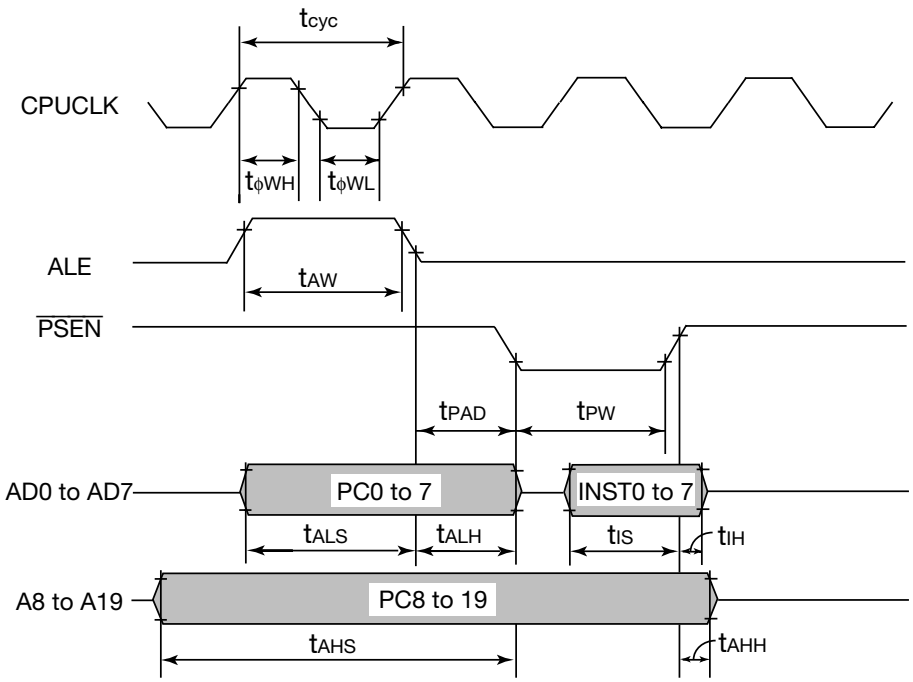
(2) Multiplexed Bus Type

1. External program memory control

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	28	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		28	—	
ALE pulse width	t_{AW}		2 tϕ – 15	—	
$\overline{PSEN}$ pulse width	t_{PW}		2 tϕ – 40	—	
$\overline{PSEN}$ pulse delay time	t_{PAD}		tϕ – 6	—	
Low address setup time	t_{ALS}		2 tϕ – 25	—	
Low address hold time	t_{ALH}		tϕ – 6	—	
High address setup time	t_{AHS}		3 tϕ – 45	—	
High address hold time	t_{AHH}		0	—	
Instruction setup time	t_{IS}		75	—	
Instruction hold time	t_{IH}		0	tϕ – 6	

Note: $t\phi = t_{cyc}/2$



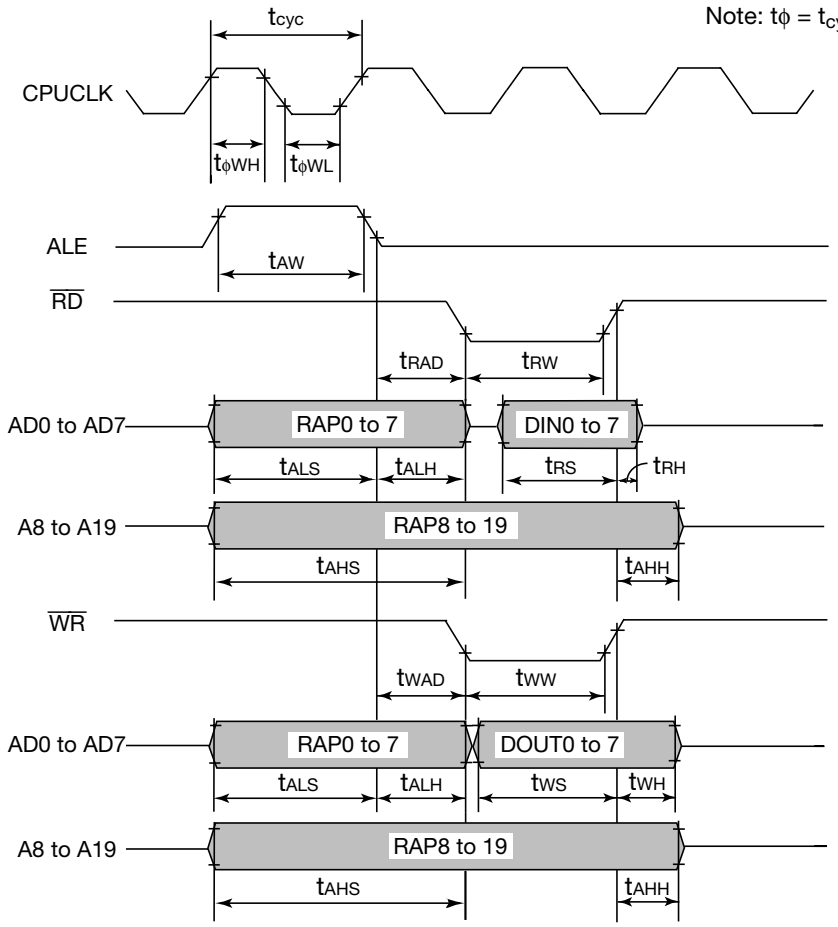
Bus timing during no wait cycle time

2. External data memory control

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	28	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		28	—	
ALE pulse width	t_{AW}		2 tϕ – 15	—	
$\overline{RD}$ pulse width	t_{RW}		2 tϕ – 40	—	
$\overline{WR}$ pulse width	t_{WW}		2 tϕ – 40	—	
$\overline{RD}$ pulse delay time	t_{RAD}		tϕ – 6	—	
$\overline{WR}$ pulse delay time	t_{WAD}		tϕ – 6	—	
Low address setup time	t_{ALS}		2 tϕ – 25	—	
Low address hold time	t_{ALH}		tϕ – 6	—	
High address setup time	t_{AHS}		3 tϕ – 45	—	
High address hold time	t_{AHH}		tϕ – 6	—	
Read data setup time	t_{RS}		75	—	
Read data hold time	t_{RH}		0	tϕ – 6	
Write data setup time	t_{WS}		2 tϕ – 55	—	
Write data hold time	t_{WH}		tϕ – 6	—	

Note: $t\phi = t_{cyc}/2$



Bus timing during no wait cycle time

(3) Serial port control

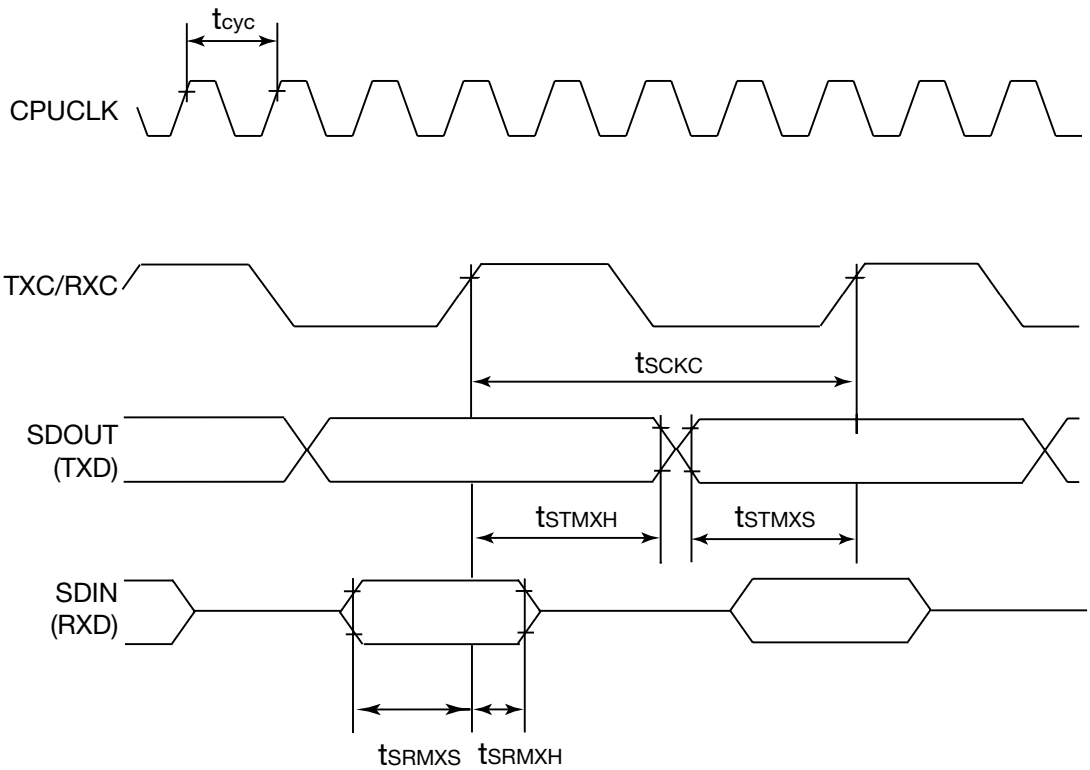
• Serial ports 1 and 6 (SIO1 and 6)

Master mode (Clock synchronous serial port)

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	4 t_{cyc}	—	
Output data setup time	t_{STMXS}		2 $t\phi$ – 10	—	
Output data hold time	t_{STMXH}		5 $t\phi$ – 20	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		0	—	

Note: $t\phi = t_{cyc}/2$

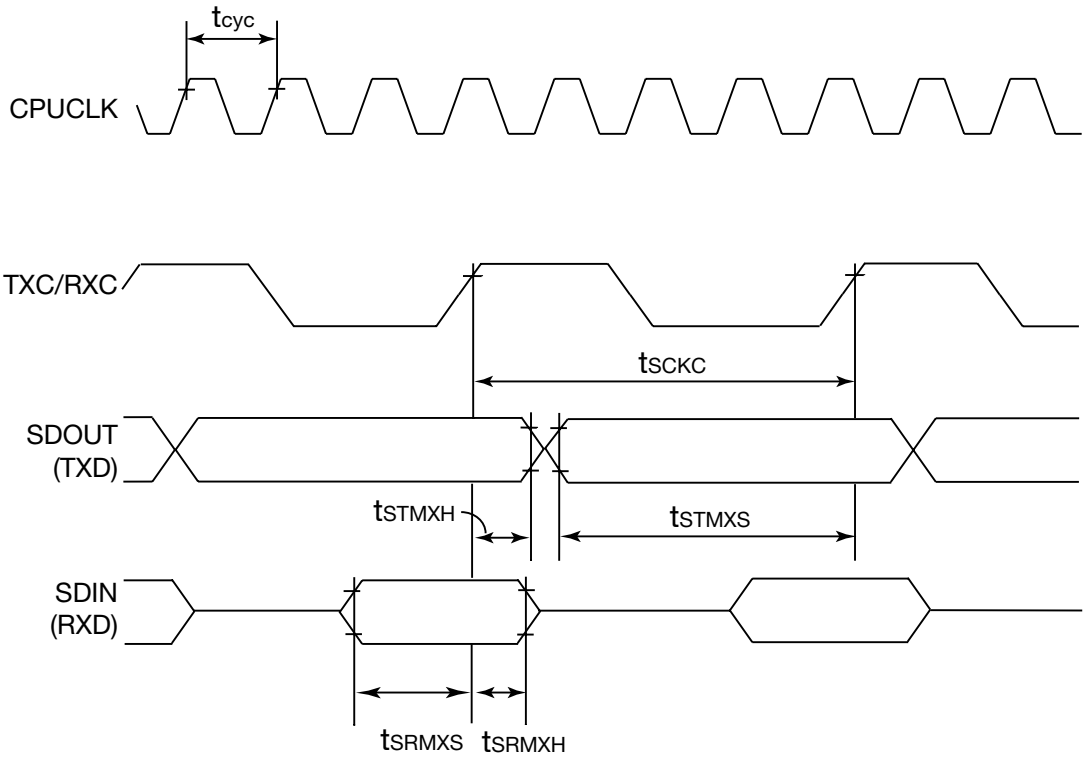


Slave mode (Clock synchronous serial port)

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	4 t_{cyc}	—	
Output data setup time	t_{STMXS}		2 tϕ – 30	—	
Output data hold time	t_{STMXH}		4 tϕ – 20	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		7	—	

Note: $t\phi = t_{cyc}/2$



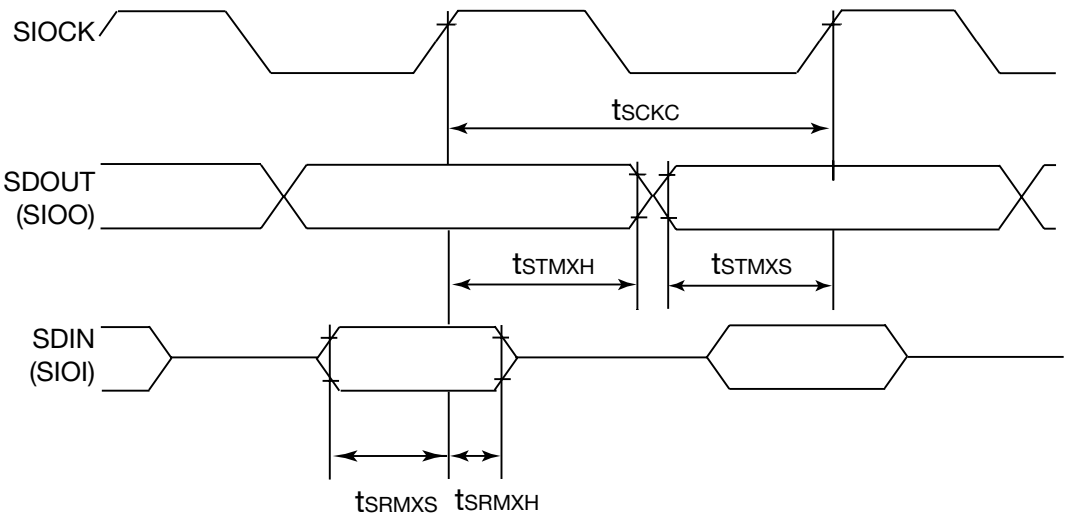
• Serial ports 4 and 5 (SIO4 and 5)

Master mode (Clock synchronous serial port)

MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	5.6 t_{cyc}	—	
Output data setup time	t_{STMXS}		5.6 $t\phi$ – 10	—	
Output data hold time	t_{STMXH}		4.2 $t\phi$ – 20	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		0	—	

Note: $t\phi = t_{cyc}/2$

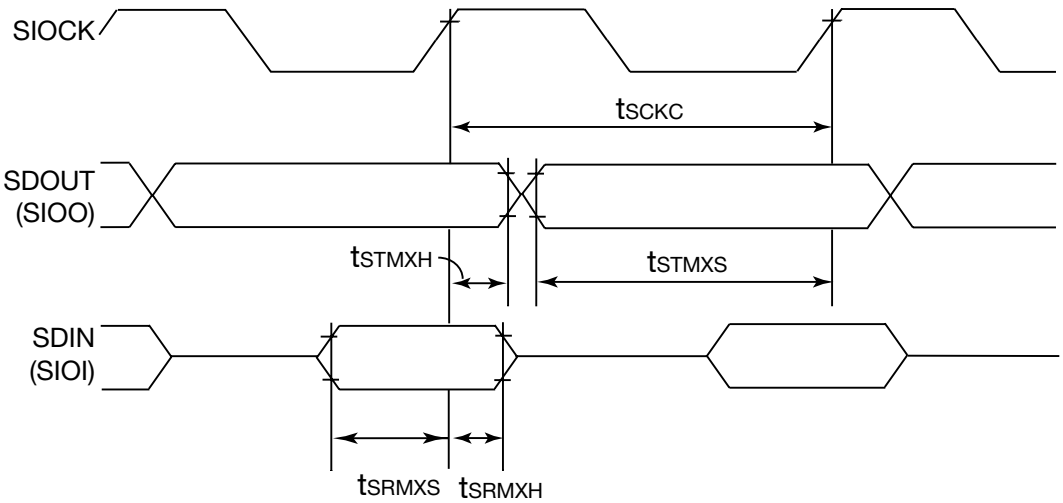


Slave mode (Clock synchronous serial port)

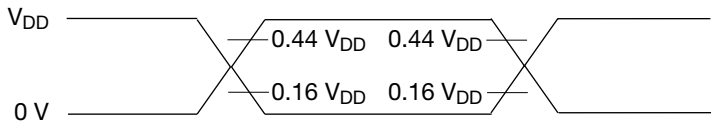
MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 14 MHz	71.4	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	5.6 t_{cyc}	—	
Output data setup time	t_{STMXS}		2.8 $t\phi$ – 30	—	
Output data hold time	t_{STMXH}		5.6 $t\phi$ – 20	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		7	—	

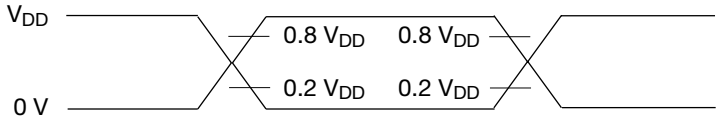
Note: $t\phi = t_{cyc}/2$



Measurement points for AC timing (except the serial port)



Measurement points for AC timing (the serial port)



20.7 A/D Converter Characteristics

20.7.1 A/D Converter Characteristics (V_{DD} = 4.5 to 5.5 V)

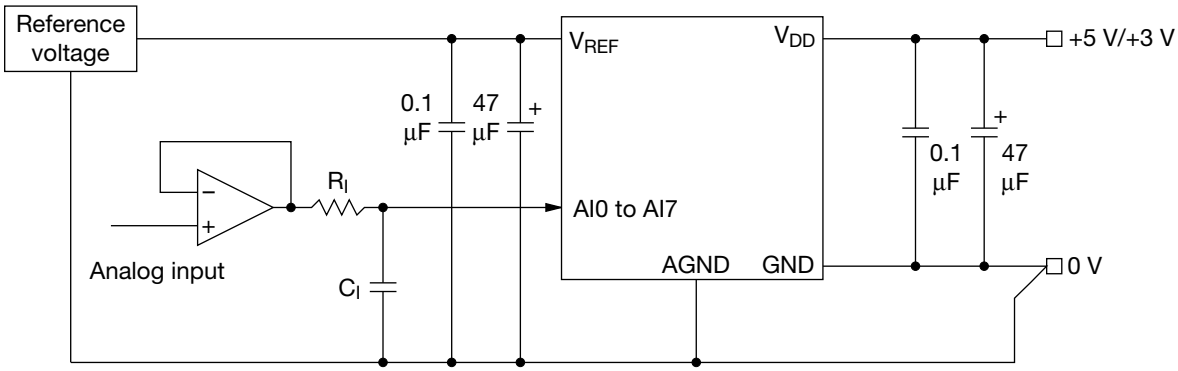
(Ta = –30 to +70°C, V_{DD} = V_{REF} = 4.5 V to 5.5 V, AGND = GND = 0 V)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	n	Refer to measurement circuit of Fig. 20-1 Analog input source impedance $R_I \leq 5$ kΩ t_{conv} = 8.5 µs	—	10	—	Bit
Linearity error	E_L		—	—	±3	LSB
Differential linearity error	E_D		—	—	±2	
Zero scale error	E_{ZS}		—	—	+3	
Full-scale error	E_{FS}		—	—	–3	
Cross talk	E_{CT}	Refer to measurement circuit of Fig. 20-2	—	—	±1	
Conversion time	t_{CONV}	Set according to ADTM set data	8.5	—	3906.3	µs/ch

20.7.2 A/D Converter Characteristics (V_{DD} = 2.4 to 3.6 V)

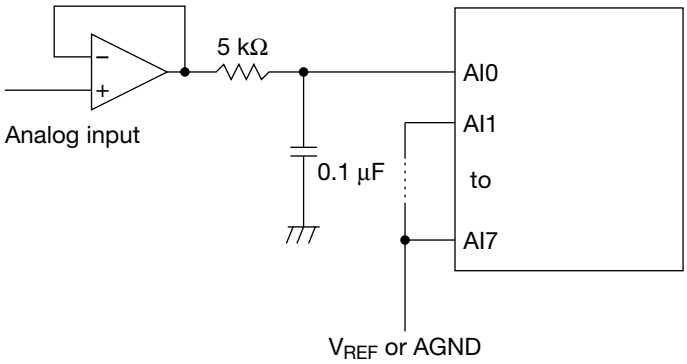
MSM66577L (Ta = –30 to +70°C, V_{DD} = V_{REF} = 2.4 to 3.6 V, AGND = GND = 0 V)
MSM66Q577LY (Ta = –30 to +70°C, V_{DD} = V_{REF} = 3.0 to 3.6 V, AGND = GND = 0 V)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	n	Refer to measurement circuit of Fig. 20-1 Analog input source impedance $R_I \leq 5$ kΩ t_{conv} = 27.4 µs	—	10	—	Bit
Linearity error	E_L		—	—	±4	LSB
Differential linearity error	E_D		—	—	±3	
Zero scale error	E_{ZS}		—	—	+4	
Full-scale error	E_{FS}		—	—	–4	
Cross talk	E_{CT}	Refer to measurement circuit of Fig. 20-2	—	—	±2	
Conversion time	t_{CONV}	Set according to ADTM set data	27.4	—	3906.3	µs/ch



R_I (impedance of analog input source) ≤ 5 kΩ
C_I ≒ 0.1 µF

Figure 20-1 Measurement Circuit



Cross talk is the difference between the A/D conversion results when the same analog input is applied to AI0 through AI7 and the A/D conversion results of the circuit to the left.

Figure 20-2 Cross Talk Measurement Circuit

Definition of Terminology

1. Resolution
   Resolution is the value of minimum discernible analog input.

   With 10 bits, since $2^{10} = 1024$, resolution of (VREF – AGND) ÷ 1024 is possible.

2. Linearity error
   Linearity error is the difference between ideal conversion characteristics and actual conversion characteristics of a 10-bit A/D converter (not including quantization error).

   Ideal conversion characteristics can be obtained by dividing the voltage between V_{REF} and AGND into 1024 equal steps.

3. Differential linearity error
   Differential linearity error indicates the smoothness of conversion characteristics. Ideally, the range of analog input voltage that corresponds to 1 converted bit of digital output is 1LSB = (V_{REF} – AGND) ÷ 1024. Differential error is the difference between this ideal bit size and bit size of an arbitrary point in the conversion range.

4. Zero scale error
   Zero scale error is the difference between ideal conversion characteristics and actual conversion characteristics at the point where the digital output changes from 000H to 001H.

5. Full-scale error
   Full-scale error is the difference between ideal conversion characteristics and actual conversion characteristics at the point where the digital output changes from 3FEH to 3FFH.

20.8 D/A Converter Characteristics

MSM66577/MSM66Q577 (V_{DD} = 2.4 to 3.6 V/4.5 to 5.5 V, Ta = –30 to +70°C)
MSM66577L (V_{DD} = 2.4 to 3.6 V, Ta = –30 to +70°C)
MSM66Q577LY (V_{DD} = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	n	—	—	—	8	Bit
Linearity error	E_L		—	—	±1	LSB
Absolute precision	—		—	—	±2	
Conversion time	t_{CONV}	C_L = 50 pF	—	20	50	µs
Analog output impedance	—	—	—	10	40	kΩ

Definition of Terminology

1. Resolution
   Resolution is the value of minimum discernible analog output.

   With 8 bits, since 2^8 = 256, resolution of (V_{DD} – GND) ÷ 256 is possible.

2. Linearity error
   Linearity error is the difference between ideal conversion characteristics and actual conversion characteristics of a 8-bit D/A converter.

   Ideal conversion characteristics can be obtained by dividing the voltage between V_{DD} and GND into 256 equal steps.

3. Absolute precision
   Absolute precision is a gross error including a linearity error and the effect of noise.

Chapter 21

Special Function Registers (SFRs)

21. Special Function Registers (SFRs)

21.1 Overview

The 256-byte area of addresses 0000H to 00FFH in data memory is the special function register (SFR) area.

The SFR area is a register group that includes special function registers such as the following.

Peripheral hardware mode registers

Arithmetic register (ACC)

Control registers (PSW, LRBL, LRBH, SSP)

21.2 List of SFRs

Table 21-1 lists the SFRs. Terms in the table are defined as follows:

- Address [H]: Addresses are expressed in hexadecimal format.
- Function: The register is named after the SFR function.
- Byte, Word:
  - Symbol: This symbol indicates an I/O function register. Each symbol indicates whether access is by byte or word.
  - —: A dash indicates that the function cannot be accessed by that unit.
- Bit Symbol:
  - Symbol: This symbol indicates an I/O function register. In some cases there are two bit symbols, however there is no difference in function.
  - Blank: The I/O register is also assigned to this bit. However, since individual access of this bit is either unnecessary or not possible, no bit symbol is needed.
  - 0: If writing to this bit, always write a value of "0". Even when writing a byte or word that includes this bit, write this bit as "0". This bit always reads as "0".

1	If writing to this bit, always write a value of "1". Even when writing a byte or word that includes this bit, write this bit as "1". This bit always reads as "1".
(1)	All writes to this bit are ignored. This bit always reads as "1".
(0)	All writes to this bit are ignored. This bit always reads as "0".
Undefined	This indicates a bit that does not support the I/O function. Programming should be done on the assumption that the value of this bit is always undefined. Operation of this bit when an in-circuit emulator is used may differ from operation when an actual chip (such as MASK or OTP versions) is used.

- R/W: This indicates whether the specified SFR can be read (R) or written (W).
  - R/W: Both read and write are possible
  - R: Read-only
  - W: Write-only
- 8/16: This indicates the unit of bit access for the specified SFR.
  - 8/16: Both 8-bit and 16-bit access are possible
  - 8: Only 8-bit access is possible
  - 16: Only 16-bit access is possible
- Initial value [H]: This indicates the contents of each SFR at reset ($\overline{RES}$ signal input, execution of a BRK instruction, overflow of the watchdog timer, or an opcode trap is generated). Values are expressed in hexadecimal format.
- Reference page: This indicates the page on which the configuration of each SFR is described.

[Note]
Do not perform the following operation on SFRs.

1. Write operations on read-only SFRs
2. Read operations on write-only SFRs
3. 16-bit operations on 8-bit SFRs
4. 8-bit operations on 16-bit SFRs
5. Operation on addresses that are not allocated as registers
6. Operations in the area for emulator use

Table 21-1 SFR List (1/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0000	System stack pointer	—	SSP									R/W	16	FF	2-23
0001		—												FF	
0002	Local register base	LRBL	LRB										8/16	Undefined	2-22
0003		LRBH												Undefined	
0004	Program status word	PSWL	PSW	MAB	F1	BCB1	BCB0	F0	SCB2	SCB1	SCB0		8/16	00	2-18
0005		PSWH		CY	ZF	HC	DD	S	F2	OV	MIE			00	
0006	Accumulator	ACCL	ACC										8/16	00	2-17
0007		ACCH												00	
0008	Table segment register	TSR	DSTSR	0	0	0	0						8/16	00	2-26
0009	Data segment register	DSR		0	0	0	0							00	2-27
000B	ROM window register	ROMWIN	—			1	1						8	Undefined	4-2
000C	ROM ready control register	ROMRDY	—	(1)	0	0	0	(1)	IRORDY	ORDY1	ORDY0		8	8B	4-4
000D	RAM ready control register	RAMRDY	—	(1)	ARDY12	ARDY11	ARDY10	(1)	ARDY02	ARDY01	ARDY00		8	FF	4-5
000E	Stop code acceptor	STPACP	—									W	8	"0"	3-3
000F	Standby control register	SBYCON	—	CLK1	CLK0	OST1	OST0	OSCS	FLT	HLT	STP	R/W	8	08	3-4
0010	Memory size acceptor	MEMSACP	—									W	8	"0"	2-2
0011	Memory size control register	MEMSCON	—	(1)	(1)	(1)	(1)	(1)	(1)	LROM	LRAM	R/W	8	FC	2-1
0013	Real-time counter control register	RTCCON	—	(1)	(1)	(1)	(1)	SELRTI1	SELRTI0	RTCIE	RTCCL		8	F0	10-2
0015	Peripheral control register	PRPHCON	—	(1)	WAIT	HOLD	EXTXT	(1)	(1)	CLKO1	CLKO0		8	8C	15-2
0018	Port 0 data register	P0	—	P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0		8	00	5-12
0019	Port 1 data register	P1	—	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0		8	00	5-14
001A	Port 2 data register	P2	—	(0)	(0)	(0)	(0)	P2_3	P2_2	P2_1	P2_0		8	00	5-16
001B	Port 3 data register	P3	—	(0)	(0)	(0)	(0)	P3_3	P3_2	P3_1	P3_0		8	00	5-18
001C	Port 4 data register	P4	—	P4_7	P4_6	P4_5	P4_4	P4_3	P4_2	P4_1	P4_0		8	00	5-20
001D	Port 5 data register	P5	—	P5_7	P5_6	P5_5	P5_4	(0)	(0)	(0)	(0)		8	00	5-22
001E	Port 6 data register	P6	—	P6_7	P6_6	P6_5	P6_4	P6_3	P6_2	P6_1	P6_0		8	00	5-24
001F	Port 7 data register	P7	—	P7_7	P7_6	(0)	(0)	(0)	(0)	(0)	(0)		8	00	5-26
0020	Port 0 mode register	P0IO	—	P0IO7	P0IO6	P0IO5	P0IO4	P0IO3	P0IO2	P0IO1	P0IO0		8	00/FF	5-12
0021	Port 1 mode register	P1IO	—	P1IO7	P1IO6	P1IO5	P1IO4	P1IO3	P1IO2	P1IO1	P1IO0		8	00/FF	5-14

Table 21-1 SFR List (2/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0022	Port 2 mode register	P2IO	—	(0)	(0)	(0)	(0)	P2IO3	P2IO2	P2IO1	P2IO0	R/W	8	00/0F	5-16
0023	Port 3 mode register	P3IO	—	(0)	(0)	(0)	(0)	P3IO3	P3IO2	P3IO1	P3IO0		8	00/02	5-18
0024	Port 4 mode register	P4IO	—	P4IO7	P4IO6	P4IO5	P4IO4	P4IO3	P4IO2	P4IO1	P4IO0		8	00/FF	5-20
0025	Port 5 mode register	P5IO	—	P5IO7	P5IO6	P5IO5	P5IO4	(0)	(0)	(0)	(0)		8	00	5-22
0026	Port 6 mode register	P6IO	—	P6IO7	P6IO6	P6IO5	P6IO4	P6IO3	P6IO2	P6IO1	P6IO0		8	00	5-24
0027	Port 7 mode register	P7IO	—	P7IO7	P7IO6	(0)	(0)	(0)	(0)	(0)	(0)		8	00	5-26
0028	Port 0 secondary function control register	P0SF	—	XAD7	XAD6	XAD5	XAD4	XAD3	XAD2	XAD1	XAD0		8	00/FF	5-12
				P0SF7	P0SF6	P0SF5	P0SF4	P0SF3	P0SF2	P0SF1	P0SF0				
0029	Port 1 secondary function control register	P1SF	—	XDM15	XDM14	XDM13	XDM12	XDM11	XDM10	XDM9	XDM8		8	00/FF	5-14
				P1SF7	P1SF6	P1SF5	P1SF4	P1SF3	P1SF2	P1SF1	P1SF0				
002A	Port 2 secondary function control register	P2SF	—	(0)	(0)	(0)	(0)	XDM19	XDM18	XDM17	XDM16		8	00/0F	5-16
								P2SF3	P2SF2	P2SF1	P2SF0				
002B	Port 3 secondary function control register	P3SF	—	(0)	(0)	(0)	(0)	WR	RD	PSEN	ALE		8	00/03	5-18
								P3SF3	P3SF2	P3SF1	P3SF0				
002C	Port 4 secondary function control register	P4SF	—	XDM7	XDM6	XDM5	XDM4	XDM3	XDM2	XDM1	XDM0		8	00/FF	5-20
				P4SF7	P4SF6	P4SF5	P4SF4	P4SF3	P4SF2	P4SF1	P4SF0				
002D	Port 5 secondary function control register	P5SF	—	P5SF7	PTM0OUT	CPCM1	CPCM0	(0)	(0)	(0)	(0)		8	00	5-22
					P5SF6	P5SF5	P5SF4								
002E	Port 6 secondary function control register	P6SF	—	PTM2OUT	P6SF6	PTM1OUT	P6SF4	P6SF3	P6SF2	P6SF1	P6SF0		8	00	5-24
				P6SF7		P6SF5									
002F	Port 7 secondary function control register	P7SF	—	PWM1OUT	PWM0OUT	(0)	(0)	(0)	(0)	(0)	(0)		8	00	5-26
				P7SF7	P7SF6										
0030	Interrupt request register 0	IRQ0	—	QCPCM1	QCPCM0	0	0	0	0	QFRCOV	QINT0		8	00	17-12
0031	Interrupt request register 1	IRQ1	—	0	QTM3OV	QTM2OV	QTM1OV	QINT3	QINT2	QINT1	QTM0OV		8	00	17-13
0032	Interrupt request register 2	IRQ2	—	QSIO1	QTM4OV	0	0	QINT7	QINT6	QINT5	QINT4		8	00	17-14
0033	Interrupt request register 3	IRQ3	—	QRTC	0	QAD	QTM6OV	QSIO4	QSIO6	QSIO5	QTM5OV		8	00	17-15
0034	Interrupt enable register 0	IE0	—	ECPCM1	ECPCM0	0	0	0	0	EFRCOV	EINT0		8	00	17-17
0035	Interrupt enable register 1	IE1	—	0	ETM3OV	ETM2OV	ETM1OV	EINT3	EINT2	EINT1	ETM0OV		8	00	17-18
0036	Interrupt enable register 2	IE2	—	ESIO1	ETM4OV	0	0	EINT7	EINT6	EINT5	EINT4		8	00	17-19
0037	Interrupt enable register 3	IE3	—	ERTC	0	EAD	ETM6OV	ESIO4	ESIO6	ESIO5	ETM5OV		8	00	17-20

Table 21-1 SFR List (3/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0038	Interrupt priority control register 0	IP0	—	0	0	0	0	P1FRCOV	P0FRCOV	P1INT0	P0INT0	R/W	8	00	17-22
0039	Interrupt priority control register 1	IP1	—	P1CPCM1	P0CPCM1	P1CPCM0	P0CPCM0	0	0	0	0		8	00	17-23
003A	Interrupt priority control register 2	IP2	—	P1INT3	P0INT3	P1INT2	P0INT2	P0INT1	P0INT1	P1TM0OV	P0TM0OV		8	00	17-24
003B	Interrupt priority control register 3	IP3	—	P1SIO0	P0SIO0	P1TM3OV	P0TM3OV	P1TM2OV	P0TM2OV	P1TM1OV	P0TM1OV		8	00	17-25
003C	Interrupt priority control register 4	IP4	—	P1INT7	P0INT7	P1INT6	P0INT6	P1INT5	P0INT5	P1INT4	P0INT4		8	00	17-26
003D	Interrupt priority control register 5	IP5	—	P1SIO1	P0SIO1	P1TM4OV	P0TM4OV	0	0	0	0		8	00	17-27
003E	Interrupt priority control register 6	IP6	—	P1SIO4	P0SIO4	P1SIO6	P0SIO6	P1SIO5	P0SIO5	P1TM5OV	P0TM5OV		8	00	17-28
003F	Interrupt priority control register 7	IP7	—	P1RTC	P0RTC	0	0	P1AD	P0AD	P1TM6OV	P0TM6OV		8	00	17-29
0040	Free running counter	—	FRC										16	00	9-3
0041		—												00	
004A	Capture compare register 0	—	CPCMR0										16	00	9-6
004B		—												00	
004C	Capture compare register 1	—	CPCMR1										16	00	9-6
004D		—												00	
0050	Free running counter control register	FRCON	—	(1)	(1)	CP1MD	CP0MD	FRRUN	FRCK2	FRCK1	FRCK0		8	C0	9-4
0051	Capture control register	CAPCON	—	(1)	(1)	CP1E1	CP1E0	CP0E1	CP0E0	(0)	(0)		8	C0	9-7
0055	Compare control register 0	CPCMCON0	—	(1)	(1)	(1)	(1)	(1)	CPCMSBF0	CPCMBF0	CPCMOUT0		8	FC	9-6
0056	Compare control register 1	CPCMCON1	—	(1)	(1)	(1)	(1)	(1)	CPCMSBF1	CPCMBF1	CPCMOUT1		8	FC	9-6
0058	External interrupt control register 0	EXI0CON	—	EX3M1	EX3M0	EX2M1	EX2M0	EX1M1	EX1M0	EX0M1	EX0M0		8	00	16-2
0059	External interrupt control register 1	EXI1CON	—	EX7M1	EX7M0	EX6M1	EX6M0	EX5M1	EX5M0	EX4M1	EX4M0		8	00	16-3
005A	External interrupt control register 2	EXI2CON	—	MIPF	NMIRD	NMIM1	NMIM0	(1)	(1)	(0)	(0)		8	0C/4C	16-4
005C	Interrupt request register 4	IRQ4	—	(1)	(1)	(1)	QTM9OV	QPWM3	QPWM2	QPWM1	QPWM0		8	E0	17-16
005D	Interrupt enable register 4	IE4	—	(1)	(1)	(1)	ETM9OV	EPWM3	EPWM2	EPWM1	EPWM0		8	E0	17-21
005E	Interrupt priority control register 8	IP8	—	P1PWM3	P0PWM3	P1PWM2	P0PWM2	P1PWM1	P0PWM1	P1PWM0	P0PWM0		8	00	17-30
005F	Interrupt priority control register 9	IP9	—	(1)	(1)	(1)	(1)	(1)	(1)	P1TM9OV	P0TM9OV		8	FC	17-31
0060	TBC clock dividing register	TBCKDVR	TBCKDV	(1)	(1)	(1)	(1)						8/16	F0	7-3
0061	TBC clock dividing counter	—		(1)	(1)	(1)	(1)					R	16	F0	7-2
0062	General-purpose 16-bit timer 0 counter	—	TM0C									R/W	16	Undefined	8-4
0063		—												Undefined	

Table 21-1 SFR List (4/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0064	General-purpose 16-bit timer 0 register	—	TM0R									R/W	16	Undefined	8-4
0065		—												Undefined	
0066	General-purpose 16-bit timer 0 control register	TM0CON	—	TM0OUT	(1)	(1)	(1)	TM0RUN	TM0C2	TM0C1	TM0C0		8	70	8-4
0068	General-purpose 8-bit timer 12 counter	TM1C	TM12C										8/16	Undefined	8-10
0069		TM2C												Undefined	8-10
006A	General-purpose 8-bit timer 12 register	TM1R	TM12R										8/16	Undefined	8-10
006B		TM2R												Undefined	8-10
006C	General-purpose 8-bit timer 1 control register	TM1CON	—	TM1OUT	(1)	(1)	(1)	TM1RUN	TM1C2	TM1C1	TM1C0		8	70	8-10
006D	General-purpose 8-bit timer 2 control register	TM2CON	—	TM2OUT	(1)	MODPWM	MOD16	TM2RUN	TM2C2	TM2C1	TM2C0		8	40	8-11
0070	General-purpose 8-bit timer 3 counter	TM3C	—										8	Undefined	8-22
0071	General-purpose 8-bit timer 3 register	TM3R	—										8	Undefined	8-22
0072	General-purpose 8-bit timer 3 control register	TM3CON	—	TM3OUT	(1)	(1)	(1)	TM3RUN	TM3C2	TM3C1	TM3C0		8	70	8-22
0074	General-purpose 8-bit timer 4 counter	TM4C	—										8	Undefined	8-28
0075	General-purpose 8-bit timer 4 register	TM4R	—										8	Undefined	8-28
0076	General-purpose 8-bit timer 4 control register	TM4CON	—	TM4OUT	(1)	(1)	(1)	TM4RUN	TM4C2	TM4C1	TM4C0		8	70	8-28
0078	General-purpose 8-bit timer 5 counter	TM5C	—										8	Undefined	8-34
0079	General-purpose 8-bit timer 5 register	TM5R	—										8	Undefined	8-34
007A	General-purpose 8-bit timer 5 control register	TM5CON	—	TM5OUT	(1)	(1)	(1)	TM5RUN	TM5C2	TM5C1	TM5C0		8	70	8-34
007C	General-purpose 8-bit timer 6 counter	TM6C	—										8	Undefined	8-43
007D	General-purpose 8-bit timer 6 register	TM6R	—										8	Undefined	8-43
007E	General-purpose 8-bit timer 6 control register	TM6CON	—	MODWDT	WDTLDE	WDTRUN	(1)	ATMRUN	WDTC2	WDTC1	WDTC0		8	10	8-41
0084	SIO1 transmit control register	ST1CON	—	TR1NIE	TR1MIE	ST1ODD	ST1PEN	ST1STB / ST1SLV	(1)	ST1LN	ST1MOD		8	04	12-4
0085	SIO1 receive control register	SR1CON	—	SR1REN	RC1IE	SR1ODD	SR1PEN	SR1SLV	S1EXC	SR1LN	SR1MOD		8	00	12-6
0086	SIO1 transmit-receive buffer register	S1BUF	—										8	Undefined	12-10
0087	SIO1 status register	S1STAT	—	(0)	(0)	RC1END	TR1END	TR1EMP	PERR1	OERR1	FERR1		8	00	12-8
008C	SIO4 control register	SIO4CON	—	(1)	BUSY4	ICK4	TEN4	SIO4SL	SIO4C2	SIO4C1	SIO4C1		8	80	12-41
008D	FIOF control register	FIFOCON	—	SRES5	ORE5	FUL5	EMP5	SRES4	ORE4	FUL4	EMP4		8	11	12-43
008E	SIO4 input FIFO data register	SIN4	—									R	8	Undefined	12-45
008F	SIO4 output FIFO data register	SOUT4	—									W	8	Undefined	12-45
0090	PWM register 0	PWR0	PWR01									R/W	8/16	00	11-4
0091	PWM register 1	PWR1												00	
0092	PWM register 2	PWR2	PWR23										8/16	00	11-4
0093	PWM register 3	PWR3												00	

Table 21-1 SFR List (5/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0094	PWM cycle register 0	PWCY0	PWCY									R/W	8/16	00	11-3
0095	PWM cycle register 1	PWCY1												00	
0096	PWM counter 0	PWC0	PWC										8/16	00	11-3
0097	PWM counter 1	PWC1												00	
0098	PWM control register 0	PWCON0	—	PWC1OV	PWCK11	PWCK10	PW1RUN	PWC0OV	PWCK01	PWCK00	PW0RUN		8	00	11-4
0099	PWM control register 1	PWCON1	—	(1)	(1)	(1)	(1)	(1)	(1)	(1)	PWHSM		8	FE	11-6
009C	A/D control register 0L	ADCON0L	—	(1)	SCNC0	SNEX0	ADRUN0	0	ADSNM02	ADSNM01	ADSNM00		8	80	13-3
009D	A/D control register 0H	ADCON0H	—	ADTM02	ADTM01	ADTM00	STS0	0	ADSTM02	ADSTM01	ADSTM00		8	80	13-5
009E	A/D interrupt control register 0	ADINT0	—	(1)	(1)	(1)	(1)	ADSTIE0	ADSNIE0	INTST0	INTSN0		8	F0	13-7
00A0	A/D result register 00	—	ADR00									R	16	Undefined	13-8
00A1		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A2	A/D result register 01	—	ADR01										16	Undefined	13-8
00A3		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A4	A/D result register 02	—	ADR02										16	Undefined	13-8
00A5		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A6	A/D result register 03	—	ADR03										16	Undefined	13-8
00A7		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A8	A/D result register 04	—	ADR04										16	Undefined	13-8
00A9		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00AA	A/D result register 05	—	ADR05										16	Undefined	13-8
00AB		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00AC	A/D result register 06	—	ADR06										16	Undefined	13-8
00AD		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00AE	A/D result register 07	—	ADR07										16	Undefined	13-8
00AF		—		(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00B8	Port 8 data register	P8	—	P8_7	P8_6	(0)	P8_4	P8_3	P8_2	P8_1	P8_0	R/W	8	00	5-28
00B9	Port 9 data register	P9	—	P9_7	(0)	(0)	(0)	P9_3	P9_2	P9_1	P9_0		8	00	5-30
00BA	Port 10 data register	P10	—	(0)	(0)	P10_5	P10_4	P10_3	(0)	(0)	(0)		8	00	5-32
00BB	Port 11 data register	P11	—	(0)	(0)	(0)	(0)	P11_3	P11_2	P11_1	P11_0		8	00	5-34

Table 21-1 SFR List (6/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
00BC	Port 12 data register	P12	—	P12_7	P12_6	P12_5	P12_4	P12_3	P12_2	P12_1	P12_0	R	8	Undefined	5-36
00BE	Port 14 mode register	P14	—	P14_7	P14_6	(0)	(0)	(0)	P14_2	P14_1	P14_0	R/W	8	00	5-37
00BF	Port 15 mode register	P15	—	(0)	(0)	(0)	(0)	P15_3	P15_2	P15_1	P15_0		8	00	5-39
00C0	Port 8 mode register	P8IO	—	P8IO7	P8IO6	(0)	P8IO4	P8IO3	P8IO2	P8IO1	P8IO0		8	00	5-28
00C1	Port 9 mode register	P9IO	—	P9IO7	(0)	(0)	(0)	P9IO3	P9IO2	P9IO1	P9IO0		8	00	5-30
00C2	Port 10 mode register	P10IO	—	(0)	(0)	P10IO5	P10IO4	P10IO3	(0)	(0)	(0)		8	00	5-32
00C3	Port 11 mode register	P11IO	—	(0)	(0)	(0)	(0)	P11IO3	P11IO2	P11IO1	P11IO0		8	00	5-34
00C4	Port 14 mode register	P14IO	—	P14IO7	P14IO6	(0)	(0)	(0)	P14IO2	P14IO1	P14IO0		8	00	5-37
00C5	Port 15 mode register	P15IO	—	(0)	(0)	(0)	(0)	P15IO3	P15IO2	P15IO1	P15IO0		8	00	5-39
00C6	Port 14 secondary function control register	P14SF	—	AO1	AO0	(0)	(0)	(0)	P14SF2	SIOO5	SIOCK5		8	00	5-37
				P14SF7	P14SF6					P14SF1	P14SF0				
00C7	Port 15 secondary function control register	P15SF	—	(0)	(0)	(0)	(0)	TXC6	RXC6	TXD6	P15SF0		8	00	5-39
								P15SF3	P15SF2	P15SF1					
00C8	Port 8 secondary function control register	P8SF	—	PWM3OUT	PWM2OUT	(0)	PTM4OUT	TXC1	RXC1	TXD1	P8SF0		8	00	5-28
				P8SF7	P8SF6		P8SF4	P8SF3	P8SF2	P8SF1					
00C9	Port 9 secondary function control register	P9SF	—	HLDACK	(0)	(0)	(0)	P9SF3	P9SF2	P9SF1	P9SF0		8	00	5-30
				P9SF7											
00CA	Port 10 secondary function control register	P10SF	—	(0)	(0)	P10SF5	SIOO4	SIOCK4	(0)	(0)	(0)		8	00	5-32
							P10SF4	P10SF3							
00CB	Port 11 secondary function control register	P11SF	—	(0)	(0)	(0)	(0)	XTOUT	CLKOUT	P11SF1	P11SF0		8	00	5-34
								P11SF3	P11SF2						
00CC	General-purpose 8-bit timer 9 counter	TM9C	—										8	Undefined	8-49
00CD	General-purpose 8-bit timer 9 register	TM9R	—										8	Undefined	8-49
00CE	General-purpose 8-bit timer 9 control register	TM9CON	—	TM9OUT	(1)	(1)	(1)	TM9RUN	TM9C2	TM9C1	TM9C0		8	70	8-49
00CF	FIFO mode register	FIFOMOD	—	(1)	(1)	(1)	(1)	(1)	(1)	FMOD1	FMOD0		8	FC	12-54

Table 21-1 SFR List (7/7)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
00D5	SIO5 control register	SIO5CON	—	(1)	BUSY5	ICK5	TEN5	SIO5SL	SIO5C2	SIO5C1	SIO5C0	R/W	8	80	12-52
00D6	SIO5 input FIFO data register	SIN5	—									R	8	Undefined	12-54
00D7	SIO5 output FIFO data register	SOUT5										W	8	Undefined	12-54
00DD	DA control register	DACON	—	(1)	(1)	(1)	(1)	(1)	(1)	(1)	DAON	R/W	8	FE	14-2
00DE	DA register 0	DAR0	—										8	00	14-2
00DF	DA register 1	DAR1											8	00	14-2
00EA	Capture compare buffer register 0	—	CPCMBFR0										16	00	9-8
00EB		—													
00EC	Capture compare buffer register 1	—	CPCMBFR1										16	00	9-8
00ED		—													
☆00F0	Flash memory acceptor	FLAACP	—									W	8	"0"	19-11
☆00F1	Flash memory control register	FLACON	—	(1)	(1)	AMPOFF	FCLK1	FCLK0	(1)	(1)	PRG	R/W	8	C6	19-12
☆00F2	Flash memory address register	—	FLAADRS	FA11	FA10	FA9	FA8	FA7	(1)	(1)	(1)		16	Undefined	19-11
☆00F3		—		(1)	(1)	(1)	FA16	FA15	FA14	FA13	FA12				
00F4	SIO6 transmit control register	ST6CON	—	TR6NIE	TR6MIE	ST6ODD	ST6PEN	ST6STB / ST6SLV	(1)	ST6LN	ST6MOD		8	04	12-16
00F5	SIO6 receive control register	SR6CON		SR6REN	RC6IE	SR6ODD	SR6PEN	SR6SLV	S6EXC	SR6LN	SR6MOD		8	00	12-18
00F6	SIO6 transmit-receive buffer register	S6BUF	—										8	Undefined	12-22
00F7	SIO6 status register	S6STAT	—	(0)	(0)	RC6END	TR6END	TR6EMP	PERR6	OERR6	FERR6		8	00	12-20

[Note]

A star (☆) in the address column indicates a SFR existing only in the MSM66Q577LY/MSM66Q577 (Flash ROM version).

For details, refer to Chapter 19, "Flash Memory".

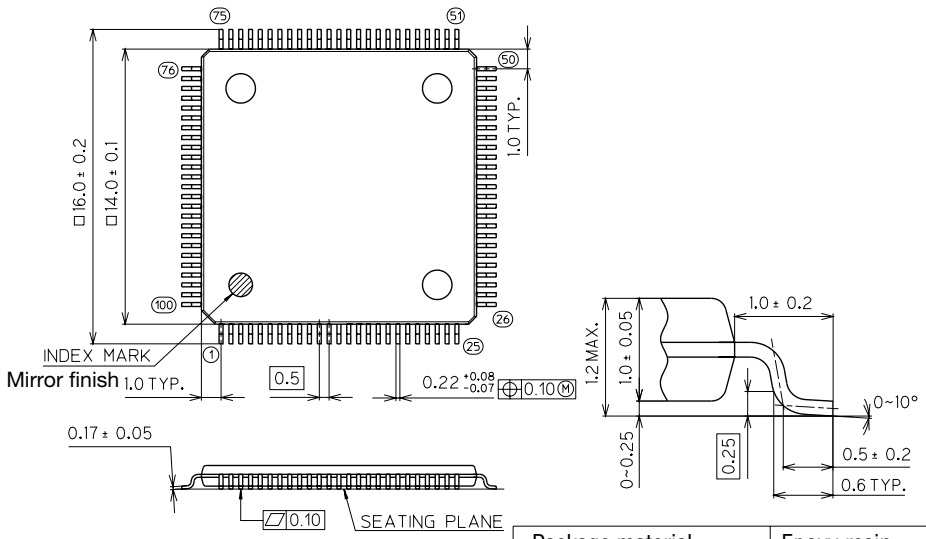
Chapter 22

Package Dimensions

22. Package Dimensions

(Unit : mm)

TQFP100-P-1414-0.50-K



Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating (≥5 μm)
Package weight (g)	0.55 TYP.
Rev. No./Last Revised	4/Oct. 28, 1996

Oki Electric Industry Co., Ltd.

Notes for Mounting the Surface Mounting Type Package

The TQFP is a surface mount type package and is very susceptible to heat in reflow mounting and to humidity absorbed in storage. Therefore, before you do reflow mounting, contact Oki's responsible sales person on the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

MSM66577 Family

User's Manual

First Edition: December 2000

PEUL66577-01